

OKTOBER 1985 ÖS 55,—/sfr. 6,50 Lit 6.200/hfl 8,—/dkr 30,— DM 6,50

10|85 DAS MAGAZIN FÜR COMPUTER-FANS

Eine Markt & Technik Publikation

Lernen

Mehr Erfolg mit dem C 64

Geld sparen durch Selbstbau

- ★ Hextastatur
- ★ Expansion-Platinen
- ★ ROM-Erweiterung

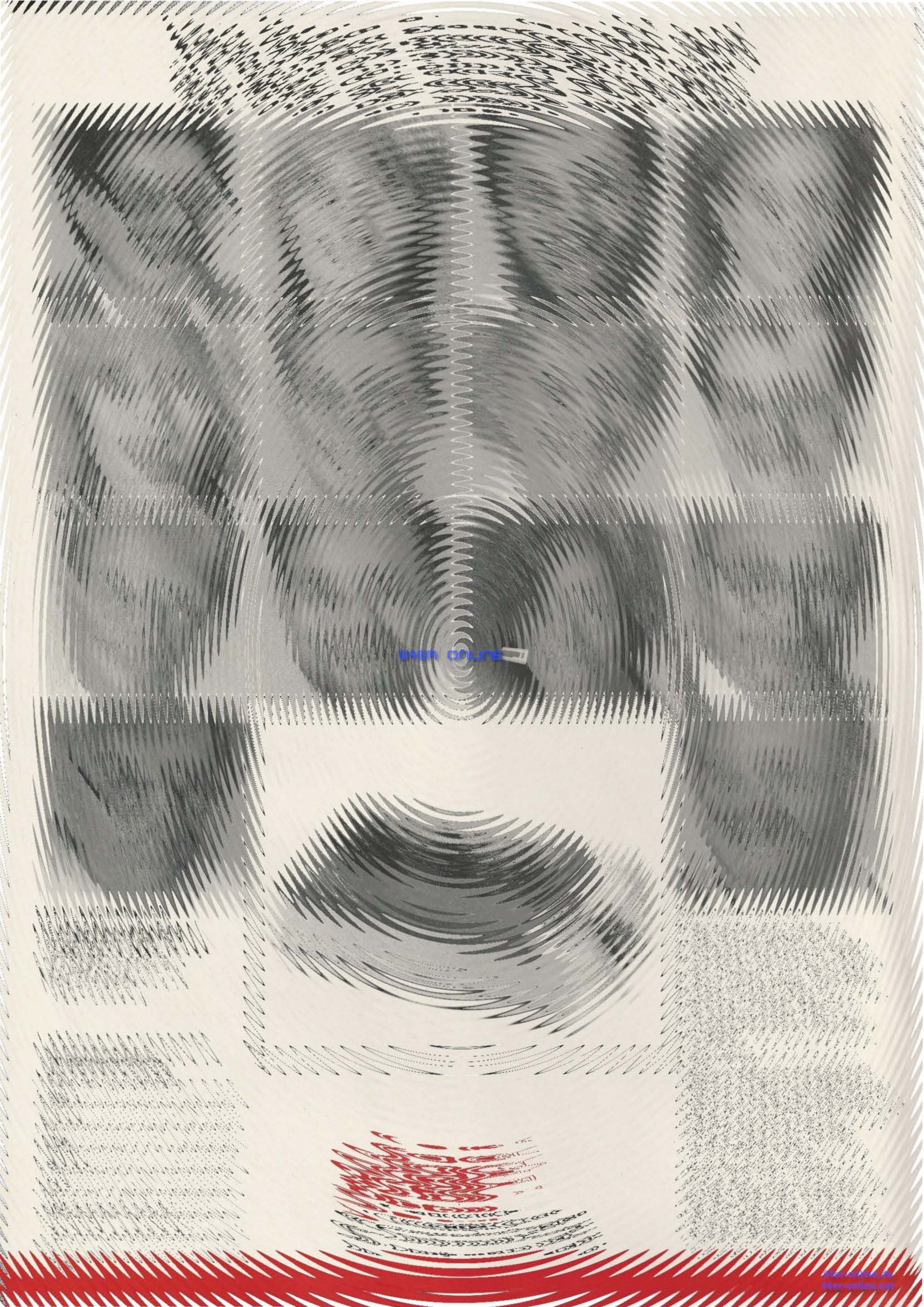
Super-Textverarbeitung zum Abtippen

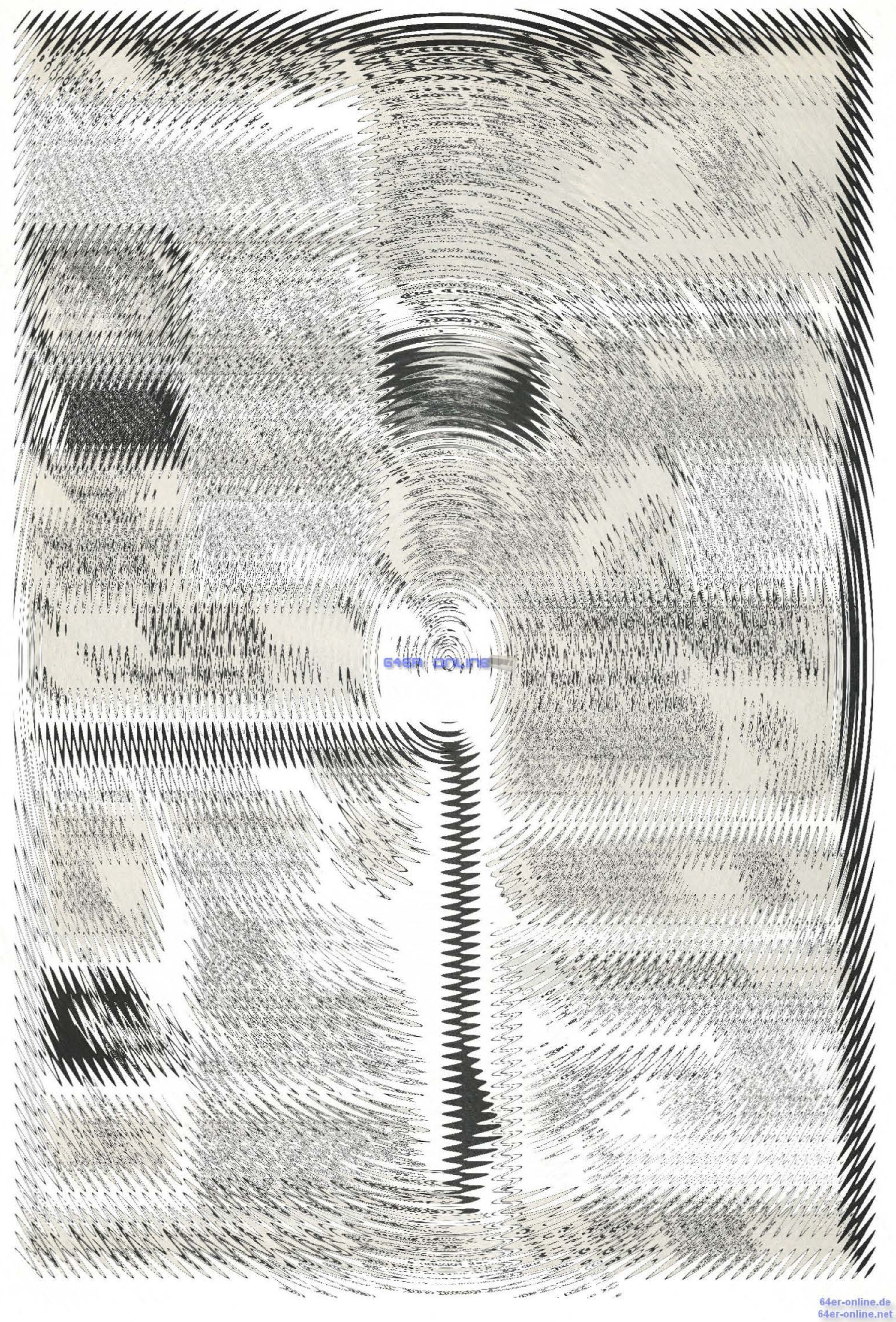
Floppy und Datasette

- ★ Schnelle Alternativen
- ★ Einstell-Anleitung
- ★ Große Disketten-Übersicht

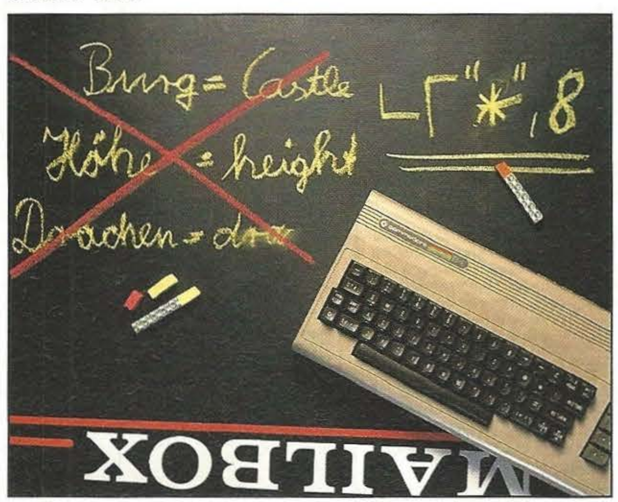
Leichter lernen mit Computer

Jetzt zusätzlich mit 128er Der Spezialteil für C128-Fans




64er online

Seite 166



Lernen mit dem Computer

Die Lernsoftware ist aus dem Stadium der Vokabellernprogramme hinausgewachsen. Lesen Sie, was Sie noch alles mit dem Computer lernen können. Zum Beispiel in Computer-Kursen. Eine ausführliche Marktübersicht zeigt Ihnen, welche Programme angeboten werden, was sie leisten und was sie kosten.
Seite 166

Seite 24



Drei Drucker im Test

Der RFI DP 165 ist ein Drucker, der bislang ein Schattendasein führte. Zu Unrecht wie unser Test ergab. Denn der DP 165 besitzt sehr gute Eigenschaften. Eine neue Konkurrenz für die Drucker-Stars. Außerdem nehmen wir für Sie Melchers CP-80 und den Epson GX-80 unter die Lupe. Beide lassen sich problemlos am C 64 anschließen.
Seite 24

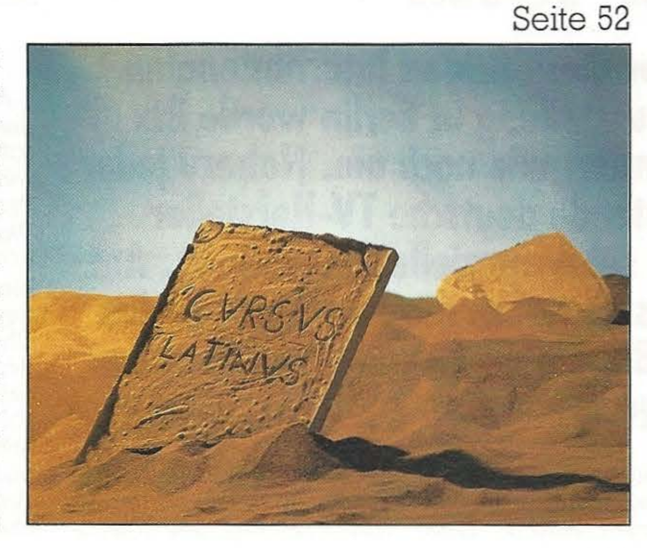
Seite 16



128er-Magazin

Sie finden eine Bauanleitung zum Anschluß eines Monitors für den 40-/80-Zeichen-Modus. Wir zeigen, was das Basic 7.0 kann.
Seite 16

Seite 52



Fit in Latein

»Cursus Latinus« ist unsere Hilfe für alle geplagten »Lateiner«. Mit »Cursus Latinus« wird das lästige Pauken von Vokabeln und Deklinationen fast zum Vergnügen. Eine klare Benutzerführung macht den Umgang mit dem Programm zum Kinderspiel. Für uns stand schon nach kurzer Testzeit fest: Das ist die Anwendung des Monats.
Seite 52

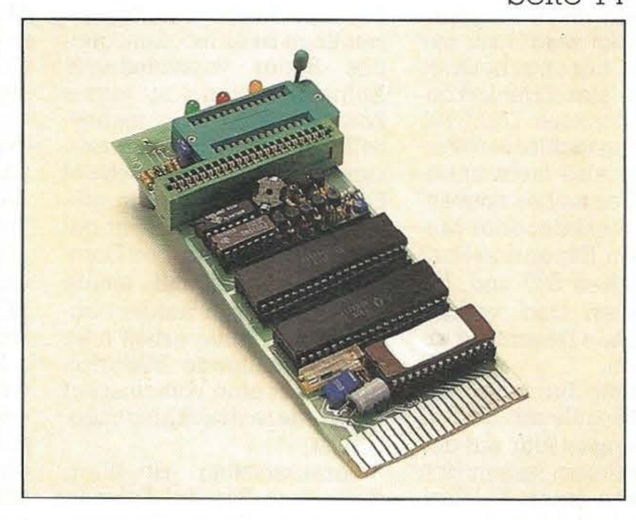
Seite 28



Massenspeicher

Zur Konservierung von Daten sind Massenspeicher ein unabdingbares »Muß«. Wir stellen Ihnen die wichtigsten Systeme vor. Darunter Kassettensysteme. Was bringen Diskettenlaufwerke mit 1 MByte Speicherplatz und Floppy-Speeder? Außerdem geben wir Ihnen Einstellhinweise für die 1541-Floppy und die Datasette.
Seite 28

Seite 14



Einer der besten Eprommer

Der Rolls-Royce unter den Eprommern: bedienungsfreundlich und komfortabel. Beides Attribute, die der Quickbyte II zu Recht verdient. Selbst ein Anfänger wird mit diesem Eprommer schnell zurechtkommen. Auch wir waren überrascht, wie einfach das Programmieren von EPROMs sein kann.
Seite 14

Listings zum Abtippen

64'er extra

Kurse

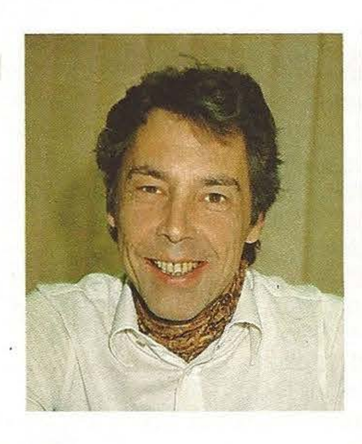
Software-Test

Spiele

Lernen mit Computer

Rubriken





Noch'n Service: Platinen

Nach dem Listing-Service, den viele Leser gern nutzen, um sich Tipparbeit zu sparen, bieten wir jetzt auch einen Platinenservice — für alle diejenigen, die sich eine Erweiterung oder Zusatzschaltung selbst bauen wollen. Die Auswertung des in Ausgabe 6 ausgeschriebenen Wettbewerbs hatte nämlich gezeigt, daß sich etwa die Hälfte aller Leser für Selbstbau interessieren — und daß ein Drittel sogar meint, das Thema sei im 64'er unterrepräsentiert. Wenn wir Bauanleitungen veröffentlichen, bieten wir künftig — erstmals ab diesem Heft — die Möglichkeit, entweder die fertige leere Platine oder einen kompletten Bausatz und/oder die fertig montierte Einheit zu beziehen. Aufbauten auf Standard-Lochrasterplatinen ergeben ja nicht immer befriedigende und selten elegante Lösungen; das Selbst-Ätzen von Platinen macht selbst routinierten Hobbyelektronikern häufig Ärger (und erfordert außerdem eine spezielle Ausrüstung). Mit der Option, auch eine fertige Einheit beziehen zu können, wollen wir denen helfen, die bisher mangels Lötkolben, Übung oder Zeit über die Selbstbauanleitungen hinwegblättern mußten. Wir wollen es unseren Lesern so leicht als möglich machen, die 64'er-Informationen schnell und einfach praktisch zu nutzen.

Michael Pauly,
Redaktions-Direktor

Btx total

Auf der diesjährigen Internationalen Funkausstellung in Berlin wurde Btx präsentiert wie noch nie. Nahezu jeder bedeutende deutsche TV-Hersteller zeigte seine spezielle Anwendung. Am interessantesten dürfte allerdings das neue Btx-Decoder-Modul von Commodore gewesen sein.

Die Teilnehmerzahlen bei Btx liegen weit hinter den Erwartungen zurück. Eine große Marktakzeptanz gerade bei den Privatanwendern ist noch nicht gegeben. Das liegt unter anderem an den relativ hohen Anschaffungskosten für ein komplettes Btx-System (Anschlußbox, Decoder, Fernseher und Eingabetastatur). Bereits vorhandene, ältere TV-Geräte können nicht oder nur unter großem Aufwand für Btx aufbereitet werden. Das soll sich mit dem Btx-Decoder-Modul für den C 64 und C 128 von Commodore ändern.

Bestückt ist dieses Modul mit dem Valvo-Chip »Eurom« und einen PAL-Coder. Damit kann jeder handelsübliche Farbfernseher mit normalen Video-Eingang eingesetzt werden. Bisher brauchten Btx-Teilnehmer einen SCART-Eingang für die Beistelldecoder. Selbst ältere Fernseher, die keine Video-Buchse haben, lassen sich anschließen. Benötigt wird dazu nur ein als Zubehör angebotener UHF-Modulator. Ebenso können die Monitore 1701/1701 und 1901 angeschlossen werden, es ist also nicht unbedingt ein Fernseher notwendig. Das Steckdecoder-Modul (für den Expansion-Port) wird zwischen 600 und 700 Mark kosten und voraussichtlich Ende Dezember erhältlich sein.

Die Vorteile durch den Einsatz des Moduls mit dem C 64/C 128 liegen klar auf der Hand: Btx-Seiten lassen sich abspeichern (spart Telefongebühren); es ist kein spezieller Fernseher notwendig (man muß nicht umsteigen); Telesoftware kann geladen werden; die Tastatur bietet alle für den Dialog erforderlichen Zeichen; der Computer ist weiterhin als solcher einsetzbar.

Die Vorteile des Computereinsatzes hat auch Nordmende erkannt. Durch ein kleines Modul kann ebenfalls der C 64 für Btx tauglich gemacht werden, allerdings ist dazu noch ein Beistelldecoder notwendig.

Eine neuartige Anwendung zeigte auch Grundig auf der IFA. Mit einem kleinen Interface lassen sich alle Grundig-Video-Recorder (sowohl System Video 2000 als auch VHS) mit IIC-Bus durch den C 64 steuern. Denkbare Einsatzgebiete sind Video-Vorführungen mit gezielter Abrufmöglichkeit, Überwachungswesen oder programmierter Unterricht.

Demonstriert wurde auf der Funkausstellung die Meteosat-Bildaufzeichnung. Durch die gute Deckung der Bildausschnitte des Wettersatelliten ist es möglich, gleiche Bilder verschiedener Aufnahmezeiten zu einer Zeitrafferfolge über mehrere Tage hinweg computergesteuert zusammenzusetzen. Die Signale werden in stehendes Fernsehbild umgewandelt und über die Computersteuerung mit einem Video-Recorder aufgezeichnet. Als Resultat erhält man eine fortlaufende Bilderfolge, mit der eine Vorschau auf die Wetterentwicklung möglich ist.

Voraussetzung ist allerdings eine Parabol-Antenne auf dem Dach! Dennoch zeigt diese Anwendung, was noch alles im C 64 steckt, man muß nur darauf kommen. Das Interface soll übrigens um die 100 Mark kosten.

Telesoftware

Auf dem Poststand wurde die Anwendung der Telesoftware demonstriert. Verwendung fanden dabei die Programme aus unserem Wettbewerb in Ausgabe 6/85. Der Sieger, Hans Liebler, erhielt aus der Hand des Bundespostministers Dr. Christian Schwarz-Schilling den 1. Preis, einen Btx-Fernseher Loewe MCP 114. Telesoftware bedeutet, daß Programme, die im Btx-Rechner gespeichert sind, von jedem Teilnehmer mit dem entsprechenden Computer (in diesem Fall der C 64) abgerufen und verarbeitet werden können. Gesucht wird allerdings noch nach einer einheitlichen Telesoftwarenorm, die es erlaubt, artfremde Daten (wie sie Programme nun einmal für Btx sind) über Bildschirmtext zu übertragen. Sollte dies in einem vertretbaren Zeitraum gelingen, dürfte Telesoftware in Btx ein zukunftsorientiertes Angebot werden.

Dazu müßte allerdings auch auf der Hardware-Seite (sprich Decoder) für den Computer ein gewisser Standard eingeführt werden. So gibt es mittlerweile neben Commodore noch mindestens zwei Anbieter von Steckdecodern für den C 64. Das Modul »Btx 64« von Astech Computer haben wir bereits in der Ausgabe 8/85, Seite 15, vorgestellt. Dieses Modul verwandelt den C 64 in eine Btx-Tastatur mit allen Computerattributen. Es kostet 248 Mark, benötigt allerdings noch einen Decoder. Wie das Decodermodul von Commodore kommt auch das SD-64 (698 Mark) von Technofor ohne Decoder aus, das heißt nahezu jeder »normale« Fernseher oder RGB-Monitor kann angeschlossen werden.

Mit einer speziellen Software (90 Mark) kann der C 64 auch als Mailbox, zum Laden von Telesoftware oder als elektronische Schreibmaschine mit Textverarbeitung eingesetzt werden. Beiden Decodern fehlt allerdings noch die Postgenehmigung.

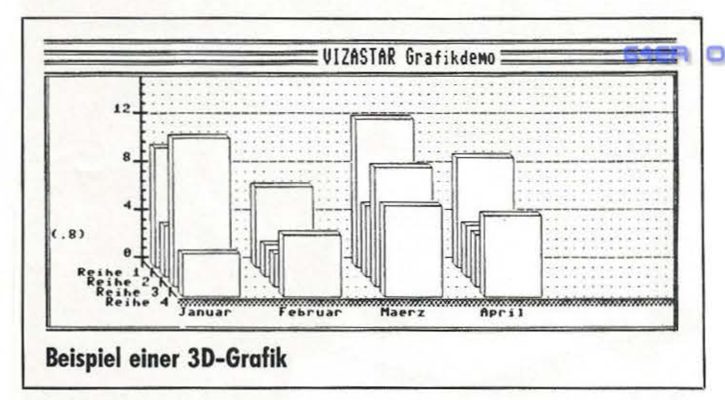
(aa)

Einen Schritt weiter — Grafiken der dritten Dimension

Extended Graphics Pack heißt die Erweiterung, mit der Vizastar 64, das integrierte Softwarepaket für den C 64, erst richtig zur Geltung kommt. Zusätzlich zu der in Vizastar 64 schon vorhandenen Linien- und Balkengrafik, fügt dieses Grafikpaket die Räumlichkeit hinzu. Statt in flächigen Balkengrafiken werden die Daten des Arbeitsblattes dreidimensional dargestellt. Jedes Säulendiagramm kann aus bis zu vier Reihen voreinander und 33 Säulen nebeneinander bestehen. Die dreidimensionale Darstellung bleibt dabei auch in den vier Reihen erhalten, denn jede Reihe wird versetzt vor der anderen abgebildet (Bild). Damit bei vielen Daten die einzelnen Säulen nicht zu klein werden, ist immer nur ein Quadrant der gesamten Grafik dargestellt. Wie schon vom Vizastar 64-Arbeitsblatt bekannt, kann man mit den Cursortasten auch über die Grafik horizontal und vertikal wandern. Jede Reihe und Spalte, Kopf- und Fußzeile kann beliebig beschriftet werden. Die zweite Darstellungsart dieser Erweiterung ist die Kreisausschnittgrafik oder einfacher Tortengrafik, sie wird allerdings nur zweidimensional abgebildet. Die Bedienung des Grafikpakets ist extrem einfach, denn die Werte für die Säulen oder Kreisausschnitte werden aus den Zellen des Vizastar-Arbeitsblatts entnommen. Auf Tastendruck wird jede Grafik in einfacher oder doppelter Größe ausgedruckt. Ganz besonders erstaunlich ist die Arbeitsgeschwindigkeit dieser Erweiterung. Ein Säulendiagramm mit vier Reihen und 33 Spalten ist bereits nach 3 Sekunden berechnet — das sind Werte, wie man sie normalerweise nicht einmal von einem Personal Computer erwarten würde.

(aw)

Info: Interface Age, Josephsburgstr. 6, 8000 München 80, Preis 75 Mark



Beispiel einer 3D-Grafik

Testen und dann erst kaufen!

Die Firma Wiesemann bietet Ihren Kunden einen neuartigen Service an: Mit dem Kauf der Interface-Typen 92000, 92000G, 92008 und 92008G erwerben Sie ein 14tägiges Rückgaberecht. Sollte Ihnen das Interface nicht gefallen, können Sie es einfach an die Firma Wiesemann zurückschicken. Sie müssen lediglich die Versand- und Nachnahmekosten tragen. Sie bekommen dann, nach Aussagen der Firma, per Überweisung Ihr Geld umgehend zurück. Abzüglich der Versandspesen. Die Rücknahmegarantie gibt nur die Firma Wiesemann selbst. Die Garantie erstreckt sich nicht auf Händler. Es ist also wichtig, daß man das Interface direkt bei Wiesemann bestellt, wenn man diesen Service in Anspruch nehmen will.

Info: Wiesemann Mikrocomputertechnik, Winchenbachstr. 3-5, 5600 Wuppertal, Tel. (0202) 505077

Neues vom Turbo Access

Die Firma Roßmüller bietet seit kurzem zwei Programme für den 1541-Floppy-Beschleuniger Turbo Access an. Es handelt sich dabei um ein menügesteuertes Kopierprogramm für einzelne Dateien und ein Disketten-Backup. Ein kurzer Test in der Redaktion brachte erstaunliche Werte. Beide Programme nutzen die Vorteile der parallelen Datenübertragung aus und kopieren einzelne Dateien oder ganze Disketten mit erstaunlicher Geschwindigkeit. Das File-Copy ist auf die Verwendung einer 1541 abgestimmt und erleichtert durch einige Sonderfunktionen die gesamte Arbeit mit dem Laufwerk. Es können DOS-Befehle gesendet und das Directory angezeigt werden. In einem »Scratch-Modus« fragt das Programm nach, welche der Dateien auf einer Diskette gelöscht werden sollen. Das

Disketten-Backup-Programm setzt das Vorhandensein von zwei Laufwerken voraus. Mit ihm lassen sich Kopien ganzer Disketten in einer Zeit von 22 Sekunden herstellen. Die Zieldiskette soll dabei gleich mitformatiert werden. Nach Auskunft des Herstellers werden von dem Kopierprogramm alle Daten zwischen der Spur 1 und der Spur 35 kopiert. Die Umgehung verschiedener Arten des Kopierschutzes soll nicht die Absicht des Programmierers gewesen sein. Beide Programme werden, nach Angabe der Firma Roßmüller, im Rahmen des Kundenservice an die Besitzer von Turbo Access auf Anforderung versendet. Gleichzeitig wurde eine Turbo-Access-Version für den C 128 angekündigt. Besitzer des alten Turbo Access benötigen dafür lediglich eine Erweiterungsplatine, deren Preis allerdings noch nicht feststand. (aw)

Info: Roßmüller GmbH, Finkenweg 1, 5309 Meckenheim 1, Tel. (02225) 14488

Kopien aus dem Automaten

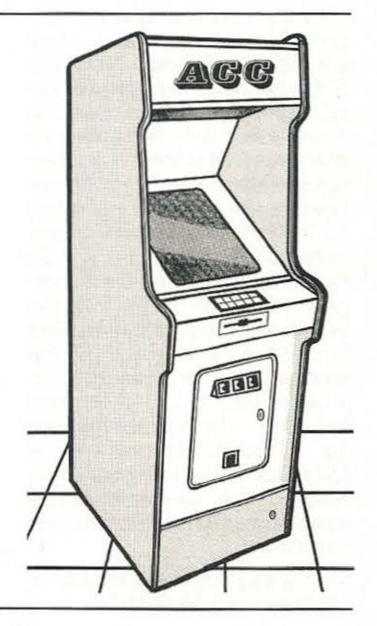
Einen neuen Verkaufsweg von Software für den C 64 will die Firma Astech Computer einschlagen: Wenn Sie ein C 64-Programm kaufen wollen, holt Ihnen der Händler keine Diskette mehr aus dem Lager. Sie müssen nur eine unformatierte Diskette in einen Automaten schieben, genügend Geld einwerfen, das Programm auswählen und in kurzer Zeit haben Sie eine Kopie des Programms in Händen. Inklusive Anleitung und Kopierschutz. Das Softwareangebot umfaßt etwa 20 Programme pro Automaten (Bild).

Jeder Kopierautomat ist über Btx mit dem Astech-Zentralcomputer verbunden. Dieser Computer fragt jeden Abend die Verkaufszahlen der einzelnen Automaten ab. Sollte sich herausstellen, daß ein Programm nicht den gewünschten Umsatz bringt, kann es innerhalb eines Tages gegen ein neues ausgetauscht werden. Ebenfalls über Btx.

Der Vorteil gegenüber den bekannten Kopierautomaten, die von einem EPROM Kopien auf Diskette herstellen, liegt in der Art der Händlerabrechnung. Ein Händler braucht nur für schon verkaufte Kopien zu bezahlen. Es besteht also nicht das Risiko, Ladenhüter einzukaufen. Außerdem kommt dem Händler der Zinsgewinn vom Verkaufserlös zugute: Einmal pro Monat bekommt der Händler die Rechnung über die verkauften Kopien.

Der Preis des Astech-Copy-Corners liegt bei 8500 Mark (netto).

Info: Astech Computer, Am Wall 183, 2800 Bremen 1, Tel. (0421) 324057





Astrologieprogramme gesucht

Ich suche Programme, beziehungsweise Algorithmen zur Astrologie, speziell auch zur Bestimmung der Gestirnstände. Gerne bin ich auch zur Zusammenarbeit mit Gleichgesinnten bereit. Elisabeth Cryns

Software für EAN-Code-Leser gesucht

Ich suche Software und Drucker, um mit meinem C 64 den Strichcode EAN 13 auf Etiketten zu drucken.

Wer hat bereits solche Software geschrieben (was ja nicht so schwierig sein kann), und wer hat bereits den C 64 in Verbindung mit EAN 13 benutzt? Bei dem Drucker sollte es sich um einen gängigen Matrix- oder Tintenstrahldrucker handeln, der auch zur Textverarbeitung genutzt werden kann.

Ralf Kammermeier

Druckbefehle bei Multiplan?

Wie kann man innerhalb des Programms »Multiplan« für den C 64 Druckersteuerzeichen (zum Beispiel für EPSON-Drucker) eingeben, um so ein etwas anderes Druckbild zu erhalten?

Thomas Prym

Undokumentierter Fehler?

Bei der Programmierung von sequentiellen Dateien gibt mir der Computer manchmal einen »FILE DATA ERROR« aus, den ich aber in keinem Handbuch finde. Welche Ursache hat dieser Fehler? Jens Richter

Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Modul-Probleme?

Ich möchte mit dem Buch »Musik auf dem Commodore 64« die Musikprogrammierung lernen. Um die Programme auf der beiliegenden Diskette laufen zu lassen benötigt man Simons-Basic, wovon ich das Modul besitze. Gleichzeitig möchte ich aber auch den JEN-Music-Computer »Musipack C 64« verwenden. Nun stehe ich vor dem Problem, jedesmal das Simons-Basic-Modul entfernen zu müssen, damit ich das Interface vom Musipack dafür einstecken kann. Dies führt jedoch auf die Dauer zu einem zu großen Verschleiß der Module.

Nun habe ich zufällig einen Adapter gesehen, mit dem man zwei Module an den C 64 anschließen kann. Nur, zu dessem Preis bekomme ich auch schon einen gebrauchten C 64. Wer weiß Rat?

Hermann Schneider

Es kommt öfter vor, daß die Preise für Zusatzhardware in keinem vernünftigen Verhältnis zu dem des C 64 stehen. Dies liegt jedoch meistens daran, daß diese Produkte (wie in diesem Fall wahrscheinlich auch) nicht sehr gefragt sind. Daran können wir jedoch leider auch nichts ändern. Sie haben daher prinzipiell zwei Möglichkeiten: Entweder Sie bauen sich den erforderlichen Adapter selber oder Sie — und dies ist in diesem Fall wohl die einfachere Lösung — besorgen sich die Diskettenversion von Simons Basic. Die muß man zwar vor jeder Benutzung laden, dafür kann aber das Musikpack-Interface im Modulschacht bleiben.

Probleme mit Reset-Taster

Ich habe vor einiger Zeit Ihren Vorschlag aus Ausgabe 7/85 »Einfacher Reset-Schalter selbst gebaut« aufgegriffen. Ich mußte jedoch leider feststellen, daß dieser Schalter nicht funktionstüchtig ist. Überarbeiten Sie bitte Ihre Pläne, da ich vermute, daß Commodore die Reset-Leitung ab Baujahr '85 nicht mehr belegt hat. Dirk Kruschewski

Commodore hat offensichtlich bei der neuesten Serie des C 64 eine kleine Änderung am seriellen Bus vorgenommen. Eine Diode am Ausgang der Reset-Leitung wirkt wie eine elektronische Einbahnstraße. Der Impuls zum Zurücksetzen der Floppy kann zwar hinaus, in umgekehrter Richtung — mittels eines kleinen Tasters an der seriellen Buchse — funktioniert der Reset jedoch nicht mehr.

Leider läßt sich nicht feststellen, bei welchem C 64 der Reset funktioniert — es hilft nur probieren.



DOS 5.1 mit Hypra-Load?

Besteht die Möglichkeit, DOS 5.1 zusammen mit »Hypra Load Perfekt« in ein EPROM zu brennen? Dieter Bast

Pearl-Compiler für C 64?

Gibt es für den C 64 einen Pearl-Compiler? Des weiteren suche ich einen Emulator und Cross-Assembler für den 6800-Prozessor. M. Volkmer

Comal-Gruppe Deutschland

Die Comal-Gruppe Deutschland hat es sich zur Aufgabe gemacht, die Verbreitung der Programmiersprache Comal nach Kräften zu unterstützen. Es steht eine große Sammlung an Programmen sowohl für die Version 0.14 als auch für das neue Comal 80 zur Verfügung. Eine eigene Clubzeitschrift bringt Kurse, Programme und Tips & Tricks. Eine Comal-Mailbox ist unter der Nummer 04683/554 zu erreichen. S. Bauer

Comal Gruppe Deutschland, S. Bauer, Kaufgasse 4, 7204 Wurmlingen, Tel. 07461/12543. Bei schriftlichen Anfragen bitte Rückumschlag beilegen.

Textverarbeitung mit Formeln

Ich besitze einen Seikosha GP-550-Drucker und suche ein Textverarbeitungsprogramm, mit dem ich mathematische Sonderzeichen ausdrucken kann. Wer hat ferner eine Hardcopy-Routine, die auch mit Simons Basic zusammenarbeitet? Ausgabe 8/85

Johann Bierschneider

Grundlage für die Darstellung mathematischer Symbole ist die Möglichkeit, eigene Steuerzeichen definieren zu können. Beim Seikosha GP-550 erreicht man das mit Hilfe des Grafikmodus. Durch geeignete Grafikdaten sollte es gelingen, das gewünschte Sonderzeichen zu drucken. Die genaue Vorgehensweise entnimmt man dem Handbuch des Druckers, insbesondere den im Anhang aufgeführten Beispielen.

Zusammenarbeit mit einem Textverarbeitungsprogramm erfolgt dann in der Regel über die Definition einer geeigneten Zahlensequenz, die angeführt von der Grafik-Einschaltsequenz (siehe Handbuch Kapitel 7) die oben ermittelten Daten enthält.

Beim Aufruf eines solchen Zahlenstrings schickt das Textverarbeitungsprogramm dem Drucker die darin enthaltenen Steuer- und Symboldaten.

Will man lediglich ein normales Zeichen hoch- oder tiefgestellt ausdrucken, schaltet man über eine Datensequenz die im GP-550 eingebaute Sub-/Superscriptschrift ein (siehe Handbuch). Frank Jörgens

C 16: Bilder auf Datasette?

Wer kann mir sagen, wie man beim C 16 fertige Bilder außer mit Shapes auf Datasette speichert? Hardeen Hornburg

Das Abspeichern von Grafiken ist sehr einfach:
Zuerst löscht man den Grafikspeicher mit »GRAPHIC 1,1«. Hat man einen Teil der Grafik fertig und will sie abspeichern, geht man mit »GRAPHIC 0,1« in den Textmodus zurück und gibt den Befehl MONITOR ein. Nun kann man die Grafik mit »S"Grafikname",1,2000,3FFF« abspeichern. Weiter geht es mit »S"Farbna-

me",1,1C00,1FFF« um die Farbe abzuspeichern und zu guter Letzt müssen auch die Luminanzwerte der Farben gesichert werden: »S"Luminanzname",1, 1800,1BFF«. Damit sind alle Komponenten der Grafik abgespeichert.

Möchte man das Bild wieder laden, dann muß zunächst der Grafikspeicher wieder gelöscht werden mit »GRAPHIC 1,1 : GRAPHIC 0,1« und anschließend geht man wieder in den Monitor. Jetzt lädt man die Files wieder in umgekehrter Reihenfolge mit »L"filename",1« in den Speicher. Hendrik Voss

Preiswerte Umschaltplatine

Die Umschaltplatine aus Ausgabe 7/85, Seite 41 (»Aus eins mach vier«) können Sie ab sofort zum Preis von 18 Mark (nur Platine) bei mir beziehen.
Andreas Gerzen

Andreas Gerzen, Marienstr. 14, 4018 Langenfeld

Programme RENUMBERn?

Können die Listings nicht vor dem Abdruck RENUMBERt werden? Dann könnte man eine automatische Zeilennumerierung benutzen. Michael Dick

Es ist technisch natürlich kein Problem, jedes BASIC-Programm vor dem Abdruck mit dem »RENUMBER«-Befehl irgendeiner Basic-Erweiterung in eine schönere Form zu bringen. Dann müßten aber auch die Zeilenangaben in den Programmbeschreibungen entsprechend abgeändert werden. Dies wäre jedoch eine neue, unnötige Fehlerquelle. Da die Zeilennummern auch ein Mittel zur übersichtlichen Programmierung sein können, setzen sie manche Programmierer zur Kennzeichnung von Unterprogrammen ein (zum Beispiel Unterprogramm 1 ab Zeile 1000, Unterprogramm 2 ab 2000 und so weiter).

Die Idee ist sehr gut, wir müssen diese Anregung jedoch an unsere Programmeinsender, mit der Bitte sie zu beherzigen, weitergeben.

Floppy-Laufwerk reinigen?

Welche Möglichkeiten habe ich, meine Floppy 1541 zu reinigen, um Lese- und Schreibfehlern vorzubeugen, und wie bekomme ich das Laufwerk 30 Sekunden lang zum Laufen, wenn ich eine Reinigungsdiskette benutzen will? Jack Reis

Sie können das Laufwerk für 30 Sekunden in Bewegung halten, wenn Sie es einfach fünfmal hintereinander initialisieren:

```
OPEN 1,8,15 : FOR I=1 TO 5 :
PRINT#1,"I" : NEXT : CLOSE 1
```

Von der übermäßigen Benutzung von Reinigungsdisketten möchten wir aber unbedingt abraten, da diese den Schreib-/Lesekopf mit der Zeit so verschleißen, daß er unbrauchbar wird. Greifen Sie daher bitte nur dann zur Reinigungsdiskette, wenn sehr häufig Lese- oder Schreibfehler auftreten und Sie sicher sind, daß dies am verschmutzten Schreib-Lesekopf liegt. Benutzen Sie Reinigungsdisketten unter keinen Umständen vorbeugend, wenn Ihnen etwas an Ihrer 1541 liegt.

Midi-Interface für C 64

Wo bekommt man ein Midi-Interface zum Anschluß des C 64 an den Synthesizer Korg Poly 61M? Mathias Heck Ausgabe 8/85

Das derzeit wohl preiswerteste Midi-Interface für den C 64 wird von der Firma Musik-Elektronik Butterweck, Hochstraße 75, 4630 Bochum 6, zum Preis von 96 Mark vertrieben. Ich benutze dieses Interface selbst und kann sagen, daß es mit praktisch allen Midi-Programmen einwandfrei zusammenarbeitet. Für den Korg Poly 61 wird keine spezielle Software angeboten, man kann aber alle Sequenzer-Programme einsetzen. Noch ein Tip: Die Zeitschrift »Keyboards« bringt in fast jeder Ausgabe Midi-Software. Reinhard Hellbach

Veröffentlichung auch ohne Drucker?

Ich würde Ihnen gerne ein Programm zur Veröffentlichung einsenden, habe aber keinen Drucker um es auszudrucken. Holger Kuhfuß

Wir freuen uns natürlich über jede Programmeinsendung, besonders wenn sie gut und ideenreich ist. Die Entscheidung, ob wir Ihr Programm abdrucken, hängt nicht vom Vorhandensein eines Listingausdrucks ab! Ausschlaggebend ist einzig und allein die Qualität des Programms und der Beschreibung. Natürlich erleichtert uns ein Listing die Arbeit mit und an Ihrem Programm. Wichtig ist jedoch, daß Sie einer solchen Einsendung einen Datenträger mit dem Programm beilegen.

Ein wichtiger Faktor bei der Beurteilung der Zusendungen ist die Güte der Anleitung, denn das beste Programm ist ohne eine gute Anleitung nichts wert. Diese sollte möglichst anschaulich sein und Ihr Programm ausführlich erläutern.

Leser fragen — Willi Brechtl antwortet

Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

Einsteiger-Literatur zum C 64 gesucht

Ich habe seit kurzem einen C 64 und möchte damit möglichst schnell programmieren lernen. Welche Bücher würden Sie mir dazu empfehlen? Annegret Stör

Das Angebot an Literatur zum C 64 ist sehr groß und speziell für den Computer-Neuling praktisch undurchschaubar. Hier ist eine Auswahl empfehlenswerter Lehrbücher für den Einsteiger:

Einen kompletten, ausführlichen Grundlehrgang in Basic gibt das Buch »Basic-Grundkurs mit dem C 64«, Markt & Technik Verlag, Haar bei München, 376 Seiten, 44 Mark.

Eine für den absoluten Neuling empfehlenswerte Einführung in den Umgang mit dem C 64 ist »64 für Einsteiger«, Verlag Data Becker, Düsseldorf, 214 Seiten, 29 Mark.

Ebenfalls an den Einsteiger wendet sich »Commodore 64 leichtgemacht«, Sybex Verlag Düsseldorf, 176 Seiten, 28 Mark.

Einen nach Basic-Befehlen geordneten, leichtverständlichen Überblick über das Commodore-Basic mit vielen Beispielen bieten gleich zwei Bücher:

Das »Commodore 64 Basic-Handbuch«, Sybex Verlag, Düsseldorf, 204 Seiten, 32 Mark, und »Basiswissen C 64«, Westermann Verlag, Braunschweig, 212 Seiten, 29,80 Mark.

Das Buch »Der Commodore 64 für Hobby, Schule und Beruf«, Carl Hanser Verlag, München, 338 Seiten, 48 Mark, bietet eine praxisorientierte, aber leichtverständliche Einführung in die Programmierung.

Anhand von Beispielen lernt man beim Buch »Programmieren — keine Ahnung?«, Birkhäuser Verlag, CH-Basel, 126 Seiten, 29,80 Mark.

Ghostbusters-Problem gelöst

Die Original-Kassette von »Ghostbusters« läuft auf meinem C 64 nicht. Woran kann das liegen? Stefan Bardos

Mit ziemlicher Wahrscheinlichkeit ist der Tonkopf Ihrer Datasette verstellt. Dies ist nicht nur allgemein die häufigste Fehlerursache bei der Datasette, sondern es reagieren auch alle Activision-Programme, wohl durch den verwendeten Kopierschutz, sehr empfindlich darauf.

Nehmen Sie also einen kleinen Schraubenzieher mit ungefähr 2 mm Klingenbreite. Entfernen Sie die Kassette aus Ihrer Datasette und drücken Sie die »Play«-Taste. Positionieren Sie den Schraubenzieher in dem kleinem Loch über der »Rewind«-Taste, in dem sich eine Schraube zur Tonkopfjustierung befindet. Drehen Sie den Schraubenzieher um etwa eine Viertelumdrehung in eine beliebige Richtung, die Sie sich aber merken müssen. Probieren Sie jetzt, das Programm zu laden. Wird der Ladevorgang jetzt früher als normalerweise abgebrochen oder erscheint gar ein »Out of Memory Error«, dann haben Sie in die falsche Richtung gedreht. Also nochmals den Schraubenzieher ansetzen, die Vierteldrehung rückgängig machen (die Richtung wissen Sie doch noch?) und jetzt eine viertel Drehung in der anderen Richtung ausprobieren. Eventuell müssen Sie mehrmals eine viertel Drehung dazugeben, bis es funktioniert. Mit dieser einfachen, wenn auch etwas zeitaufwendigen Methode bekommen Sie die richtige Einstellung Ihrer Datasette in fast allen Fällen heraus.

Thomas Staudt



Quickbyte II — das Kraftpaket

Der Quickbyte II läutet eine neue Epoche der EPROM-Programmierung ein. Selbst Neulinge auf diesem Gebiet können, dank der hervorragenden Bedienungsfreundlichkeit, schon bald ihre eigenen EPROM-Module herstellen.

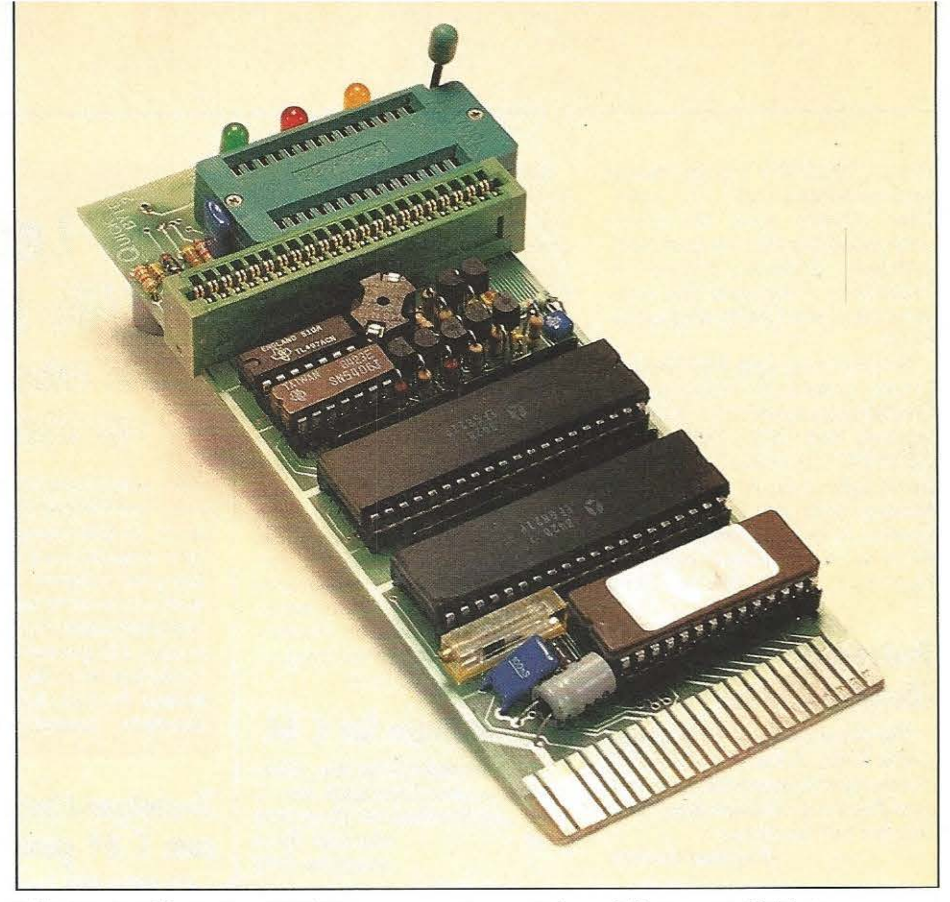


Bild. Der Quickbyte II — EPROMs programmieren wird zur Selbstverständlichkeit

Vergleicht man die Fähigkeiten heutiger EPROM-Programmiergeräte mit denen der ersten Generation, so mutet das wie ein Vergleich zwischen einem Doppeldecker und einer Concorde an. Bei den ersten Geräten war es noch Usus, alle notwendigen Programmierspannungen und EPROM-Typen (es waren meistens nur zwei oder drei) mit Schaltern und Hebelchen einzustellen. Ein EPROM-Brenner gehört heute aber schon fast zur Grundausstattung vieler Computer-Besitzer.

Das einzige, was beim Quickbyte II (siehe Bild) noch an die Pioniertage der EPROM-Programmierung erinnert, ist der offene Blick auf die sauber aufgebaute Platine (nach Aussage des Herstellers soll ein Gehäuse in Zukunft mitgeliefert werden). Alle anderen Funktionen der Hardware muten sehr professionell an. Der Aufwand, der zur Gewinnung der Programmierspannung getrieben wurde, ist hoch, denn der Quickbyte II wird am Expansion-Port (durchgeschleift) des C 64 angeschlossen. Mit dem Anschluß am Expansion-Port ist natürlich der Vorteil verbunden, die Treibersoftware ebenfalls auf dem EPROM-Brenner, in Form eines Autostart-EPROMs, unterbringen zu können. Gleich nach dem Einstecken des Quickbyte II meldet sich diese Software mit einem Menü, das schon beim ersten Durchlesen der Unterpunkte einen Vorgeschmack auf die Leistungsfähigkeit dieses Gerätes gibt. Kaum hat man die Punkte des Hauptmenüs durchgelesen, schaltet Quickbyte II selbständig in ein zweites Menü um und fordert zur Wahl eines bestimmten EPROM-Typs auf. Zu seinem Repertoire gehören die in der Tabelle dargestellten Typen. Besonders bemerkenswert ist, daß auch elektrisch löschbare EPROMs und die von Commodore verwendeten Firmware-ROMs (im Computer/Floppy/Drucker eingebaut) zu den lesbaren, Typen gehören. Kaum ein EPROM-Brenner, der ohne eingebauten Monitor ausgeliefert wird. Beim Quickbyte II hat dieser Monitor aber professionelle Qualitäten (Tabelle). Ebenso der Modul-Modus, der es erlaubt, bis zu 128 KByte lange EPROM-Auto-Start-Programme herzustellen. Auch der in mehrfacher Hinsicht intelligente Programmiermodus, der mit den immer noch recht teuren EPROMs sehr schonend umgeht, hebt den Quickbyte II aus dem weiten Feld der Konkurrenten heraus. Der Quickbyte II unterscheidet zwischen drei Programmierzeiten, die er sich, je nach Qualität des eingesetzten EPROMs, selbst sucht. Alle leeren Bytes (FF) werden nicht mitprogrammiert, sondern übersprungen (Urzustand eines EPROMs). Da sofort nach dem Schreiben ein Verivy durchgeführt wird, werden fehlerhafte Bytes unmittelbar erkannt. Der Quickbyte II erhöht dann selbständig die Programmierzeit etwas und brennt das betreffende Byte nochmals. Dadurch wird ein optimaler Kompromiß zwischen Schonung des EPROMs, Geschwindigkeit und Datensicherheit erreicht.

Mit seiner üppigen Funktionsausstattung und dem gelungenen Hardware- und Software-Konzept ist der Quickbyte II sicherlich einer der besten EPROM-Brenner für den C 64. Der Preis von 298 Mark ist für dieses Kraftpaket gerechtfertigt, auch wenn es wünschenswert wäre, ihn preiswerter zu erhalten. (aw)

- — Fast Load eingebaut (von Diskette)
- — Textool-Sockel
- — Lesen, schreiben, vergleichen, Leertest
- — Menüführung in allen Punkten
- — Auslesen von Steckmodulen
- — Programmierung einzelner Bytes
- — Diskettenoperationen
- — Module bis 128 KByte erstellen
- — intelligenter, schneller Algorithmus
- — Monitor mit vielen Funktionen
- \+ R Registeranzeige
- \+ G Sprung zu Adresse
- \+ M Speicherinhalte ansehen
- \+ L Laden von Programmen
- \+ S Speichern von Programmen
- \+ Rechenoperationen (+,-,$,#)
- \+ F Füllen von Speicherbereichen
- \+ T Speicherbereiche verschieben
- \+ H Suchfunktion
- \+ D Disassemblieren
- \+ ES,EC Spritesdefinition
- — EPROM-Typen: 27916/27513/27512/27256/57256/27128/27128/A/2764/27C64/87C64/2732/2732A/2716/2564/2532/2516/2508/5133/5143/XL2816A/XL2864A/XL48C64/2332/2364

Tabelle. Die Leistungen des Quickbyte II

Info: Jann Datentechnik, Glimmerweg 22, 1000 Berlin 47, 298 Mark



Ein Monitor ist genug

Der C 128 benötigt eigentlich zwei Monitore: einen mit RGB- und einen mit Composite-Eingang. Mit dieser Bauanleitung für eine automatische Signalumschaltung reicht allerdings ein Monitor aus, wenn Sie im 80-Zeichen-Modus auf Farbe verzichten können.

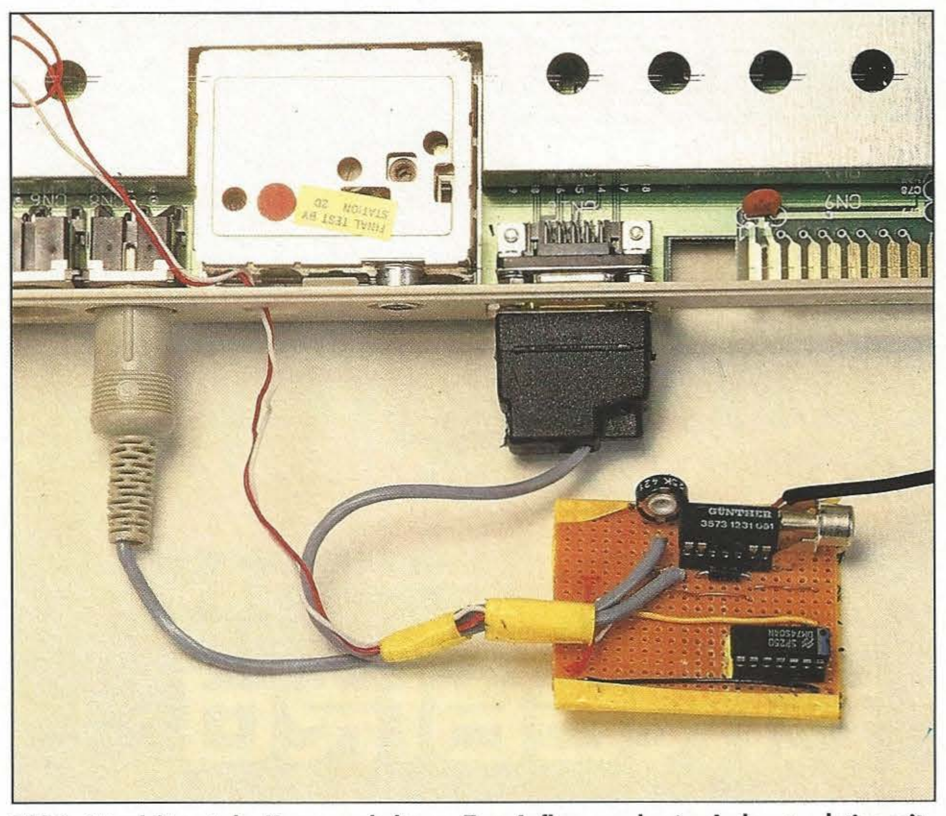


Bild 2. Die elektronische Normumschaltung. Zum Aufbau wurde eine Lochrasterplatine mit Streifen verwendet.

Kaum hatten wir die ersten C 128 in der Redaktion stehen, ärgerten wir uns über das lästige Umstecken des SW-Monitors. Composite-Ausgang im 40-Zeichenmodus, RGB im 80-Zeichenmodus. »So nicht!«, dachten wir und überlegten uns eine Schaltung, die Ihnen und uns in Zukunft die ewige Stöpselei erspart.

Schaut man sich die Belegung der beiden Video-Ausgänge des C 128 genauer an, fällt auf, daß beide Buchsen einen Luminanz(Helligkeits)-Ausgang haben. Beim RGB-Ausgang wird das Luminanz(Helligkeits)-Signal im Handbuch nur als Monochromsignal bezeichnet.

Mit einem Luminanzsignal kann jeder gebräuchliche SW-Monitor angesteuert werden. Auch der 1701/1702-Monitor von Commodore hat einen Luminanzeingang. Bei den letzteren kann auch der FBAS-Eingang an der Frontseite mit einem Luminanzsignal beschaltet werden. Man muß dann nur den Farbregler auf Schwarz-weiß drehen.

Die einfachste Methode, ein und denselben Monitor sowohl an den RGB- als auch an den Composite-Ausgang anzuschließen, ist die Verwendung eines Adaptersteckers. Dazu wird an den RGB-Ausgang ein kurzes Zwischenkabel mit RGB-Stecker und Composite-Buchse angeschlossen. Bei der 80-Zeichendarstellung muß der Monitor an das Zwischenkabel angeschlossen werden, bei 40-Zeichendarstellung und C 64-Modus an den Composite-Ausgang des C 128.

Das bringt allerdings einige Probleme mit sich. Erstens wird das Umstecken schnell lästig und zweitens ist der Kontaktverschleiß an den Steckern und Buchsen sehr hoch. Es muß also eine automatische, verschleißfreie Umschaltung her.

Automatische Umschaltung, aber wie? Über dieses Problem haben wir uns Gedanken gemacht. Wir wollten Ihnen eine optimale Lösung anbieten, die einfach nachgebaut werden kann. Nach einigen Ideen wie User-Port-Schaltungen oder Logik-Bausteinen an der MMU und den beiden Videoteilen kamen uns der richtige Einfall: die 40/80-Taste. Diese Taste, die direkt auf den Betriebsmodus Einfluß nimmt, müßte doch für die Umschaltung verwendbar sein. Tatsächlich wirkt diese Taste, ein 1xEIN-Schalter (kein Taster), auf den Pin 48 der MMU. Dies ist der Eingang, über den die MMU den 40- oder 80-Zeichen-Modus nach dem Einschalten oder nach einem Reset initialisiert. Im Grundzustand hat dieser Eingang logischen High-Pegel.

Ist der 40/80-Zeichen-Eingang der MMU unbeschaltet, wird beim Initialisieren der 40-Zeichen-Modus aktiviert, andernfalls der 80-Zeichen-Modus. Die 40/80-Taste schaltet den MMU-Eingang auf Masse, wenn der 80-Zeichen-Modus aktiviert werden soll. Man braucht also nur die Leitung vom Schalter zur MMU anzapfen, und schon ist man anhand des Logikpegels über den Darstellungsmodus nach dem Einschalten oder einem Reset informiert. 0V bedeutet 80-Zeichen-Modus, +5V zeigt die 40-Zeichendarstellung an. Da die Anschlüsse des 40/80-Schalters aus der Grundplatte herausragen, ist das »Anzapfen« ein leichtes: Das Schaltkabel für die Umschaltelektronik muß nur an den Pin des 40/80-Zeichenschalters angelötet werden, der der C 128-Rückseite zugewandt ist.

Automatische Umschaltung

Die Umschaltung der Luminanzsignale vom RGB- und Composite-Ausgang erfolgt einfach über ein kleines Reed-Relais (1xUM) in einem DIL-Gehäuse. Diese Relais brauchen bei 5V Schaltspannung einen Schaltstrom von etwa 10 bis 20 mA. Zuviel für eine direkte Ansteuerung mit dem MMU-Eingang. Der Pegel vom MMU-Eingang muß also verstärkt werden. Ein TTL-LS-Hex-Inverter ist billig und besitzt eine ausreichende »Verstärkung«. Der Low-Power-Schottky-Typ sollte nicht

durch einen normalen TTL-Baustein ersetzt werden, da der mehr Versorgungsstrom benötigt und einen kleineren Eingangswiderstand besitzt.

Die Umschaltplatine wird über Pin 2 des Kassetten-Ports mit +5V versorgt (siehe Handbuch). Es sind also nur zwei Drähte anzulöten. Verwenden Sie dazu am besten Schaltlitze, die knickfester als Draht ist. Der Lötkolben sollte eine Leistung von 16 Watt haben und gut vorgeheizt sein. Als Lötzinn eignet sich nur sogenanntes Elektroniklot.

Die Schaltung umfaßt nur wenige Teile (Schaltplan, Bild 1). Der Aufbau sollte deshalb nicht zu schwer sein. Haben Sie die Schaltung fertig aufgebaut (Bild 2), sollten Sie diese vor dem Anschluß an den Computer mit einer 4,5 Volt Batterie überprüfen. Schließen Sie dazu Masse an den Minuspol der Batterie und +5V an den Pluspol der Batterie an. Wenn Sie nun die Steuerleitung an den Minuspol der Batterie legen, sollten Sie ein leises Klicken des DIL-Relais hören. Haben Sie nur einen Ersatztyp des angegebenen Relais bekommen, lassen Sie sich unbedingt die Anschlußbelegung davon zeigen. Der Diodentyp ist unkritisch. Die Diode dient nur zum Abfangen der Induktionsspannung, die beim Abschalten des Relais auftritt. Die Polung der Induktionsspannung ist der angelegten Spannung entgegengesetzt.

Ist die Schaltung soweit in Ordnung, können Sie die Stecker anschließen. Verwenden Sie dazu einadriges, abgeschirmtes Kabel. Die Abschirmung wird nur mit Pin 2 des Composite-Steckers verbunden und nicht mit dem Steckergehäuse. Entsprechend mit Pin 1 des RGB-Steckers. Die Steckerbelegungen im Schaltplan zeigen die Lötseiten. Hier die genaue Belegung der Videobuchsen des C 128:

RGB-Ausgang

Pin	Signal	Pegel	Impedanz
1	Masse	0V	—
2	Masse	0V	—
3	Rot	0/5V	75 Ohm
4	Grün	0/5V	75 Ohm
5	Blau	0/5V	75 Ohm
6	Intensität	0/5V	75 Ohm
7	Luminanz	0-3V_{SS}	75 Ohm
8	Horiz. Synch.	0/5V	75 Ohm
9	Vert. Synch.	0/5V	75 Ohm

Composite-Ausgang

Pin	Signal	Pegel	Impedanz
1	Lumin./Synch.	1V_{SS}	75 Ohm
2	Masse	0V	—
3	Audio-Ausg.	1V_{SS}	—
4	Composite	1V_{SS}	75 Ohm
5	Audio-Eing.		—
6	Chrominanz	1V_{SS}	75 Ohm

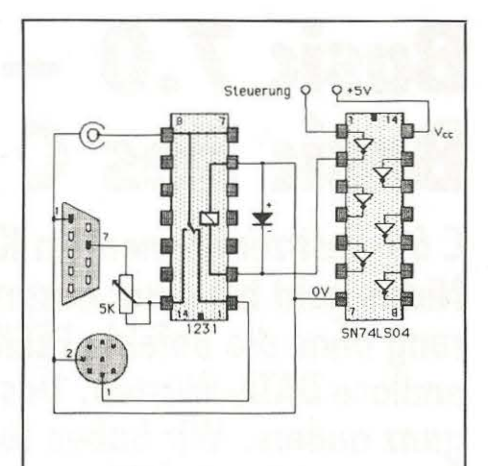


Bild 1. Der Schaltplan. Bei den Steckern werden die Lötseiten gezeigt.

Für den Monitoranschluß ist eine Cinch-Buchse zum Aufbau vorgesehen. Die Buchse wird einfach auf der Platine angelötet. Der Außenkontakt wird mit Masse verbunden, der Innenleiter mit dem Signal.

Vor der Inbetriebnahme sollten Sie noch einmal alle Anschlüsse genau überprüfen. Ist alles in Ordnung, schließen Sie das Steuerkabel und die Spannungsversorgung an den C 128 an. Verbinden Sie den SW-Monitor mit der Cinchbuchse und die beiden Videostecker mit dem C 128. Dann können Sie den Computer endlich einschalten. Wenn Sie alles richtig angeschlossen haben, erscheint die Einschaltmeldung auf dem Bildschirm. Entweder im 40- oder 80-Zeichen-Modus, je nachdem, wie die 40/80-Taste geschaltet ist. Erscheint im 80-Zeichenmodus kein Bild, drehen Sie das Trimmpoti und den Helligkeitsregler des Monitors voll auf.

Der Trimmer dient zur Abschwächung des Luminanzsignals vom RGB-Ausgang, da dieses Signal stärker als das des Chrominanz-Ausgangs ist. Der Abgleich ist sehr einfach:

1. C 128 einschalten
2. Mit ESC X den 40-Zeichenmodus einschalten (40/80-Taste entrasten)
3. Am Monitor Helligkeit und Kontrast für 40-Zeichen-Modus einstellen
4. Mit ESC X 80-Zeichenmodus einschalten und 40/80-Taste drücken
5. Mit dem Trimmpoti die Helligkeit auf den 40-Zeichenmodus anpassen.

In der gleichen Weise können Sie die Helligkeit des 80-Zeichenmodus an die des C 64-Modus anpassen. Sie müssen nur immer die 40/80-Zeichentaste umschalten, wenn softwaremäßig zwischen 80 und 40 Zeichen pro Zeile umgeschaltet wird.

Farbe ist auch möglich!

Der Composite-Ausgang bietet neben dem Luminanz- und Chrominanzsignal noch ein komplettes Video-Signal an (FBAS, gemischtes Farb- und Helligkeitssignal). Mit diesem Signal kann jeder Farbmonitor mit Videoeingang angesteuert werden. Beim Commodore-Monitor 1701/1702 ist dieser Eingang an der Frontseite und kann mit einem Schalter an der Rückseite aktiviert werden. An diesen Eingang können Sie beim 1701/1702 übrigens auch das Luminanzsignal des RGB-Ausgang legen. Eventuelle Farbverschiebungen lassen sich beseitigen, indem man den Farbregler einfach auf SW dreht.

Um das FBAS-Signal auszunutzen, schließen Sie den automatischen Umschalter nicht an Pin 1 der Composite-Buchse an, sondern an Pin 4 (unterhalb Pin 1). Wenn Sie dann den Fronteingang (Einschalten!) des 1701/1702-Monitors mit der Cinchbuchse verbinden, erfolgt die 40-Zeichendarstellung (C 128, C 64) in Farbe. Der 80-Zeichen-Modus bleibt Schwarz-weiß. Der Nachteil dieser Lösung liegt in der schlechten Auflösung des 1701/1702. 80 Zeichen pro Zeile sind kaum noch zu entziffern.

Für die meisten SW-Monitore ist das FBAS-Signal nicht geeignet. Häufig stören dann Bildstreifen die Lesbarkeit.

Für welche der beiden Lösungen Sie sich auch entscheiden, bauen Sie auf jeden Fall die Schaltung in ein kleines Gehäuse ein. Nur so ist gesichert, daß kein Kurzschluß durch herumliegende Metallteile entsteht. (hm)

1 74LS04
1 DIL-Reed-Relais 1xUM, Typ 1231 (Fa. Günther)
1 Trimmpoti 5 KOhm
1 Diode 4002 oä.
2 IC-Fassungen 14polig
1 Cinch-Platinen-Buchse
1 Kleingeräte-Stecker, 8polig, 270 Grad
1 Cannon-Stecker, 9polig mit Gehäuse
1 Stück Lochrasterplatine mit Kupferstreifen, etwas Schaltdraht und Litze
1 m einadriges abgeschirmtes Kabel
1 Gehäuse

Kosten: etwa 17 bis 20 Mark ohne Gehäuse

Stückliste

Basic 7.0 – Das Super-Basic des C 128

C 64-Besitzer können ein Klagelied davon singen: Nichts geht bei einer anspruchsvolleren Programmierung ohne die Befehle PEEK und POKE, ohne SYS und endlose DATA-Wüsten. Das wird nun mit dem C 128 ganz anders. Wir haben für Sie dieses völlig neue Programmier-Gefühl ausprobiert.

Schwachstellen im System sind die Vorliebe vieler Programmierer. Sie überlegen tagtäglich, wie man die Floppy schneller, das Druckbild schöner und das Basic komfortabler machen kann. Gerade letzteres eröffnet dem C 64-Benutzer ein großes Betätigungsfeld. Mit dem C 128 und dessen Basic 7.0 ändert sich das nun grundlegend. Viele Funktionen, die auf dem C 64 mühselig in Maschinensprache realisiert werden mußten, stehen beim C 128 als komfortable Basic-Befehle zur Verfügung. Harte Zeiten also für Maschinensprache-Freaks? Auf jeden Fall aber rosige Zeiten für Basic-Programmierer. Auf den nächsten Seiten wollen wir Ihnen zeigen, wie einfach nun das Programmieren geworden ist.

Strukturierte Programmierung

Einige Befehle des 7.0-Basic erlauben strukturierte Programmierung. So mancher Anwender wird sich nun fragen, welchen Nutzen man eigentlich davon hat. Nun, durch die Struktur wird ein Programm wesentlich übersichtlicher, das heißt man kann die Arbeitsweise wesentlich leichter durchschauen und verstehen, als die des normalerweise verwendeten »Spaghetticodes«, bei dem alle Verzweigungen durch GOTO-Befehle realisiert werden.

Nun aber zu den neuen Befehlen im einzelnen. Die DO..LOOP-Schleife, ist in etwa mit den FOR..NEXT-Befehlen vergleichbar. Damit ist es unter anderem möglich, GOTO-freie Schleifen zu erzeugen. Der Programmteil zwischen DO und LOOP wird endlos wiederholt. Damit sich der Computer jedoch nicht in dieser Schleife verfängt, können auch Bedingungen angegeben werden, unter denen die Schleife ausgeführt wird.

WHILE und UNTIL sind in ihrer Arbeitsweise sehr ähnlich. Sie können entweder dem DO-Kommando (also zum Beispiel DO WHILE) folgen, oder hinter dem Schleifenrumpf stehen (LOOP UNTIL). WHILE übersetzt man sinvollerweise mit »arbeite, solange die Bedingung erfüllt ist«. Die Schleife wird also dann beendet, wenn die Bedingung **nicht** mehr erfüllt ist:

```
10 DO WHILE A<=5
20 : INPUT A
30 LOOP
40 PRINT "ENDE"
```

In diesem kleinen Beispiel wurde eine Schleife programmiert, die so lange durchlaufen wird, bis man eine Zahl eingibt, die größer als 5 ist. Dann ist die Schleifenbedingung nicht mehr erfüllt und die Programmausführung wird in Zeile 40 fortgesetzt.

UNTIL ist genau das Gegenstück zum WHILE-Kommando. Hier gilt: Die Abarbeitung wird beendet, sobald die Bedingung erfüllt ist (das Basic bleibt also in der Schleife, solange die Bedingung nicht erfüllt ist). Aber auch hierzu wieder ein kleines Beispiel:

```
10 DO
20 : INPUT A
30 LOOP UNTIL A<5
40 PRINT "ENDE"
```

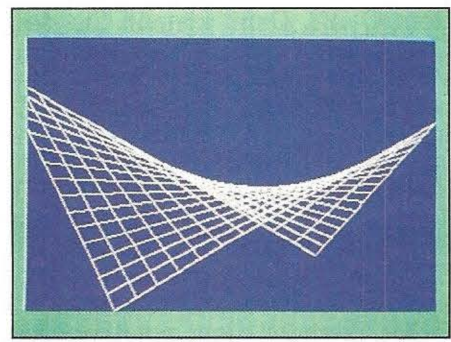


Bild 1: Dank der neuen Befehle kann man solche Figuren ohne viel Aufwand zeichnen

Diesmal arbeitet das Programm genau anders herum, das heißt wenn der eingegebene Wert von A kleiner als 5 ist, dann verzweigt das Programm nach Zeile 40. Welchen der beiden Befehle man einsetzt ist Geschmackssache, denn man kann die Bedingung immer so »verdrehen«, daß beide verwendet werden können.

Die Aufmerksamen unter Ihnen werden bestimmt schon bemerkt haben, daß die Bedingung im ersten Beispiel beim DO, im zweiten aber bei LOOP steht. Steht UNTIL oder WHILE am Schleifenkopf, dann fragt das Basic bereits beim Eintritt in die Schleife, ob die Aussprungbedingung zutrifft. Dies würde wiederum beim eben gezeigten Beispiel dazu führen, daß der Computer den Programmablauf gleich nach der Eingabe von »RUN« wieder beenden würde, da die Variable A dann ja die Null enthält. Damit wäre die Bedingung nicht erfüllt, obwohl man noch gar keine Gelegenheit gehabt hatte, in Zeile 20 einen Wert einzugeben.

Eine weitere Möglichkeit, eine Schleifenkonstruktion zu verlassen, bietet der Befehl EXIT. Wie der Name schon sagt, beendet man damit die laufende Abarbeitung und verzweigt in die nächste Zeile nach LOOP. In der Regel wird man den EXIT-Befehl daher von einer Bedingung abhängig machen.

```
100 REM ************************************
110 REM ***    HYPERBOLISCHES PARABOLID   ***

130 REM ************************************
140 REM
145 GRAPHIC1,1
150 SCALE 1,1023,1023
155 COLOR 1,2
160 A= 800:B= 300:C= 200:D= 700:X= 15
170 FOR T= 0 TO X: DRAW 1,B/X*T,C/X*T+D/X*(X-T)
    TO A+B/X*T,(C+D)/X*T: NEXT
180 FOR T= 0 TO X: DRAW 1,A/X*T,D/X*(X-T)
    TO B+A/X*T,C+D/X*T: NEXT
```

Listing 1. Das Basic-Programm zu Bild 1

Eine weitere Quelle von Ärgernissen beim Programmieren waren bisher die IF..THEN-Konstruktionen. Hinter dem THEN fanden oft nur drei bis vier Befehle Platz, so daß man oftmals umständlich im Programm hin- und herspringen mußte, um die Ausführung fortzusetzen. Auch dies wurde jetzt mit den Kommandos BEGIN und BEND anders.

Man setzt in den THEN-Zweig einfach ein BEGIN-Kommando ein. Trifft die Bedingung bei der IF..THEN-Abfrage zu, dann behandelt das Basic alle folgenden Zeilen bis BEND so, als würden sie in der gleichen Zeile hinter dem THEN-Befehl stehen. Erst ein BEND-Befehl

beendet die Abarbeitung der THEN-Folge.

Ebenfalls neu ist beim 7.0-Basic das Befehlswort ELSE. Findet der Computer im Basic-Programm eine IF-Abfrage vor, dann hat er prinzipiell zwei Möglichkeiten. Entweder die Bedingung trifft zu, dann wird der Programmteil nach THEN ausgeführt, der sich jetzt dank BEGIN und BEND über beliebig viele Zeilen erstrecken kann. Andernfalls ignoriert das Basic den Rest der Zeile und fährt in der nächsten mit der Programmabarbeitung fort. Genau für diesen Fall würde der Befehl ELSE vorgesehen. Wie er eingesetzt wird, soll wiederum ein Beispiel verdeutlichen:

```
10 INPUT A
20 IF A=2 THEN B=9 : ELSE B=2
30 PRINT B
```

Der ELSE-Zweig wird in der Abfrage genau dann ausgeführt, wenn die Bedingung (hier A = 2) nicht zutrifft. Die Benutzung dieser Konstruktion hat wiederum den Vorteil, daß man nicht mehr um die nächste Zeile »herumspringen« muß, was wiederum der Übersichtlichkeit des Programms zugute kommt.

Auch im ELSE-Zweig können die neuen BEGIN- und BEND-Kommandos eingesetzt werden, wodurch sich dieser wiederum auf mehrere Zeilen ausdehnen läßt.

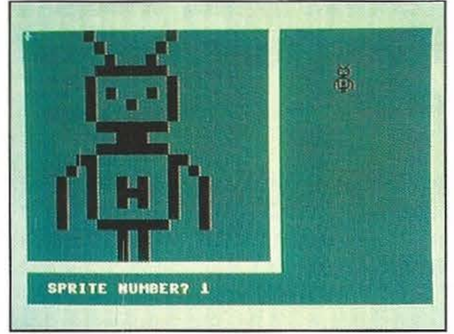


Bild 2: Der eingebaute Sprite-Editor des C 128

Auch auf einem ganz anderem Gebiet hat sich beim 7.0 Basic einiges verändert. Gegenüber dem Basic 2.0 des C 64 sind sehr komfortable Kommandos zum Zeichnen von Punkten, Linien, Rechtecken und Kreisen hinzugekommen. Auch das Ausfüllen von geschlossenen Flächen ist mit Basic 7.0 kein Problem mehr. Leider beziehen sich alle Grafik-Befehle nur auf die bereits vom C 64 bekannte Grafikauflösung von 320 x 200 Punkten. Die ebenfalls — wenn auch auf Umwegen — mögliche doppelte Auflösung von 640 x 200 Punkten wird vom Basic leider nicht unterstützt. Die Grafik-Befehle des 7.0-Basic entsprechen den bereits vom 3.5-Basic des C 16 und Plus 4 bekannten Kommandos. Auch die zahlreichen Befehlserweiterungen für den C 64 verwenden ähnliche Grafik-Kommandos. In Listing 1 ist ein kleines Beispielprogramm zu finden, um den Lesern, die keine C 64-Basic-Erweiterung besitzen, zu demonstrieren, wie einfach selbst kompliziertere Figuren (Bild 1) erstellt werden können.

Wirklich neu sind hingegen die Kommandos für die Verwaltung von Sprites. Beginnen wir mit der Definition eines solchen beweglichen Objekts. Sehr hilfreich ist da der eingebaute Sprite-Editor, der entweder vom Programm aus oder direkt mit »SPRDEF« aufgerufen werden kann (Bild 2). Hierin wird man zunächst einmal nach der Nummer des Sprites gefragt. Hat man diese Zahl zwischen 1 und 8 eingegeben, so kann es mit dem eigentlichen Sprite-Entwurf losgehen. Dazu bewegt man den Zeichenzeiger mit den normalen Cursortasten auf dem 24 x 21 großen Feld hin und her. Das Löschen beziehungsweise Setzen von einzelnen Pixels erfolgt dann mittels der Tasten 1 und 2 (bei Multicolorsprites auch noch die 3 und 4). Ist der künstlerische Entwurf geglückt, dann kann man mit SHIFT-RETURN die Definition beenden.

```
10 PRINT"HABEN SIE IHRE SPRITES
   SCHON DEFINIERT ?": GETKEY A$
20 IF A$= "J" THEN 50
30 PRINT"SPRITE #1": SLEEP 1: SPRDEF
40 PRINT"SPRITE #2": SLEEP 1: SPRDEF
50 COLLISION 1,1000
60 SCNCLR
70 SPRITE 1,1,1,0,1,1
80 SPRITE 2,1,2,1,1,1
90 MOVSPR 1,300,100
100 MOVSPR 2,0,100
110 MOVSPR 2,90#4
115 DO
120 IF BUMB (1)=0 THEN SCNCLR
130 LOOP
1000 SCNCLR: CHAR 1,10,20,"BOOOOOMMM"
1010 RETURN
```

Listing 2. So einfach ist der Umgang mit Sprites in Basic 7.0

Mit dem Editieren allein ist es natürlich nicht getan. Darum werden vom 7.0-Basic auch Befehle für das Setzen und Bewegen von Sprites zur Verfügung gestellt. Das Beispiel in Listing 2 soll hier wiederum die neuen Kommandos verstehen helfen. Hier hat der Benutzer zunächst einmal die Möglichkeit, sich zwei Sprites zu erstellen. Dazu wird der eingebaute Editor zweimal aufgerufen.

Der nächste neue Befehl COLLISION legt eine Programmzeile fest, in die das Basic verzweigen soll, wenn sich zwei Objekte treffen. Besonders positiv ist, daß sich der Benutzer nicht mehr explizit um die Abfrage zu kümmern braucht. Hat man die Ansprungzeile einmal festgelegt (in unserem Beispiel Zeile 1000), so unterbricht das Basic bei einer Kollision der entsprechenden Sprites automatisch den Programmablauf und verzweigt in die vorgegebene Zeile. Dort kann der Programmierer dann angemessen auf das Ereignis reagieren. Ist diese Abarbeitung beendet, so kehrt man mittels »RETURN« wieder in das zuvor unterbrochene Programm zurück.

Nun aber wieder zurück zu unserem Beispielprogramm in Listing 2. Als nächstes müssen die Attribute für jedes Sprite wie zum Beispiel Sichtbarkeit, Vordergrundfarbe, Priorität und so weiter, angegeben werden. Diese Festlegung erledigt man über den SPRITE-Befehl. Die Positionierung und das Bewegen der Objekte, wird mit einem anderen Kommando bewerkstelligt: mit »MOVSPR 1,200, 300« positioniert man Sprite 1 auf der Bildschirmposition X = 200 und Y = 300. Mit »MOVSPR 1,90 # 5« bringt man es dann auch noch zum Laufen. Die 90 gibt dabei die Himmelsrichtung an (90 = Bewegung nach rechts), die 5 ist für die Geschwindigkeit zuständig. Diese Bewegungen werden per Interrupt gesteuert, daß heißt auch darum muß sich der Benutzer nicht kümmern. Ist der Bewegungsablauf einmal programmiert, geht alles Weitere automatisch. Schaltet man die Sprites nicht wieder ab, so bewegen sie sich sogar nach Beendigung des Programms weiter über den Bildschirm!

Speichern und Laden von Sprites ist im neuen Basic natürlich auch möglich. Der Befehl SPRSAV überträgt die Grafikdaten — je nach Stellung der Parameter — in Strings oder zurück. Dies ist sehr einfach: Gibt man »SPRSAV 1, A$« ein, so wird das Sprite in diesem String abgespeichert. Vertauscht man die Parameter, also »SPRSAV A$,1«, so erreicht man den umgekehrten Prozeß: Der Inhalt des Strings A$ wird als Bitmuster dem Sprite Nummer 1 zugewiesen.

Dies war nun eine erste kleine Einführung in die Befehlsfülle des neuen 7.0-Basic. Mit einer näheren Beschreibung aller Basic-Befehle wäre leicht ein ganzes Buch zu füllen. Aber auch die Beschreibung der interessantesten neuen Sprachelemente dürfte schon so manchem C 64-Besitzer einen Vorgeschmack darauf geben, welche neuen Programmier-Dimensionen sich mit dem Basic 7.0 auftun.

(Christoph Sauer/ev)

FRAGEN und ANTWORTEN zum 128er

Der neue Commodore 128 hat ein großes Echo unter unseren Lesern gefunden. Trotz ausführlicher Berichterstattung tauchen immer wieder neue Fragen auf, die zum C 128 gestellt werden. Die am häufigsten gestellten Fragen werden hier beantwortet.

Welche Bedeutung haben die vielen beim C 128 vorhandenen Sondertasten?

Die vier allgemeinen Funktionstasten (F1 bis F8) oberhalb des Ziffernblocks haben eine ähnliche Aufgabe wie beim C 64. Allerdings sind im C 128-Modus diese Tasten bereits vorbelegt:

F1	GRAPHIC	Grafik-Modus ein
F2	DLOAD"	Programm laden
F3	DIRECTORY	Directory anzeigen
F4	SCNCLR	Bildschirm löschen
F5	DSAVE"	Programm speichern
F6	RUN	Programm starten
F7	LIST	Programm listen
F8	MONITOR	Maschinensprache-Monitor aktivieren

Man kann die Belegung dieser Tasten mit dem KEY-Kommando anzeigen oder ändern.

Links neben den Funktionstasten befinden sich vier zusätzliche Cursortasten, die in ihrer Funktion mit den rechts unten vorhandenen Cursortasten übereinstimmen (nicht wirksam im C 64-Modus).

Die restlichen Sondertasten haben spezielle Funktionen:

Mit der ESC-Taste können spezielle ESC-Funktionen ausgelöst werden (Tabelle 1).

TAB ist eine Tabulator-Taste, mit der der Cursor an die nächste Tabulatorposition gesetzt wird.

Mit der ALT-Taste können Sonderfunktionen beliebigen anderen Tasten zugeordnet werden. Das ist allerdings vom verwendeten Programm abhängig. Soll eine Funktion aufgerufen werden, so ist die zugeordnete Taste zusammen mit der ALT-Taste zu drücken.

Die ASCII/DIN-Taste schaltet zwischen ASCII- und DIN-Zeichensatz und der entsprechenden Tastaturbelegung um. Sie funktioniert auch im C 64-Modus (!), allerdings wird hier nur der Zeichensatz geändert, während die Tastaturbelegung gleich bleibt. Das führt zu recht merkwürdigen Ergebnissen. Zum Beispiel erreicht man im C 64-Modus den Umlaut »Ä« durch die Tastenkombination (COMMODORE)-F. Bild 1 zeigt die komplette Tastaturbelegung im C 64-Modus, wenn die DIN-Taste aktiviert ist. Die ASCII/DIN-Taste ist die einzige Sonderfunktions-Taste, die auch im C 64-Modus eine Wirkung hat.

Die HELP-Taste dient zum leichten Auffinden von Fehlern in Basic-Programmen. Sie sollte immer dann gedrückt werden, wenn das Programm aufgrund einer Fehlermeldung abbricht. Dadurch wird die Zeile, in der der Fehler auftrat, am Bildschirm aufgelistet und die Fehlerstelle revers markiert.

LINE FEED erzeugt auf dem Bildschirm einen Zeilenvorschub ohne Wagenrücklauf (Carriage Return). Wird diese Taste gedrückt, wandert der Cursor um eine Zeile nach unten, ohne seine derzeitige Spaltenposition zu verlassen.





Bild 1. Tastaturbelegung des C 128 im C 64-Modus bei gedrückter DIN-Taste

Pin	Signal
1	Masse
2	Masse
3	Rot
4	Grün
5	Blau
6	Intensität
7	Monochrom-Signal
8	Horizontale Synchronisation
9	Vertikale Synchronisation

40/80 DISP schaltet vom 40- zum 80-Zeichen-Bildschirm um. Diese Taste rastet beim Drücken ein und muß bereits vor dem Einschalten des Computers in die gewünschte Position gebracht werden. Eine Betätigung bei laufendem Computer bleibt bis zu einem Reset wirkungslos.

NO SCROLL verhindert das Bildschirmscrolling. Wird diese Taste beispielsweise während des Listens eines Programms gedrückt, dann wird die Auflistung nach 25 Zeilen unterbrochen und erst auf Tastendruck fortgesetzt.

Wie schließt man einen beliebigen RGB-Monitor an den C 128 an?

Falls der Monitor über einen Standard-RGB-Stecker verfügt, gibt es keine Probleme: Einfach den Stecker in die RGB-Buchse des C 128 einstöpseln, und der Anschluß ist getätigt. Andernfalls finden Sie in Bild 2 die Pin-Belegung der RGB-Buchse am C 128. Eine Zeichnung der Buchse finden Sie auch im Commodore-Handbuch, Anhang L, Seite 1 (Steckerbelegungen). Leider wurde im Handbuch aber vergessen, die dazugehörige Pin-Belegung abzudrucken. Mit den Informationen aus Bild 2 sollte sich jeder RGB-Monitor anschließen lassen. Das Monochrom-Signal entspricht dem Luminanz-Signal bei der Composite-Video-Schnittstelle. Über dieses Signal ist auch der Anschluß eines monochromen Monitors möglich.

Taste	Funktion
A	Insert-Modus ein
B	Untere rechte Ecke eines Windows definieren
C	Insert-Modus aus
D	Bildschirmzeile löschen
E	Cursor-Blinkmodus aus
F	Cursor-Blinkmodus ein
G	Akustisches Signal ein
H	Akustisches Signal aus
I	Neue Bildschirmzeile einfügen
J	Cursor an Zeilenanfang setzen
K	Cursor an Zeilenende setzen
L	Bildschirm-Scrolling ein
M	Bildschirm-Scrolling aus
N	Normal-Modus 80-Zeichen-Bildschirm
O	Insert-, Anführungs- und Invers-Modus aus
P	Bildschirmzeile bis Cursor-Position löschen
Q	Bildschirmzeile ab Cursor-Position löschen
R	Invers-Modus 80-Zeichen-Bildschirm
S	Block-Cursor ein
T	Obere linke Ecke eines Windows definieren
U	Strich-Cursor ein (nur bei 80 Zeichen)
V	Rollt Bildschirm um eine Zeile nach oben
W	Rollt Bildschirm um eine Zeile nach unten
X	Umschaltung 40/80 Zeichen und zurück
Y	Voreingestellte Tabulatorstops setzen
Z	Alle Tabulatorstops löschen
@	Bildschirm ab Cursor-Position löschen

Tabelle 1. Die ESC-Funktionen beim C 128

Bild 2. Pinbelegung der RGB-Buchse am C 128

Welche Grafik-Modi stehen beim C 128 zur Verfügung?

Der C 128 verfügt über fünf verschiedene Grafik-Modi, die über den GRAPHIC-Befehl ausgewählt werden können:

GRAPHIC 0: Text 40 Zeichen pro Zeile
GRAPHIC 1: Hochauflösende Grafik
GRAPHIC 2: Hochauflösung mit Textfenster
GRAPHIC 3: Multicolor-Grafik
GRAPHIC 4: Multicolor mit Textfenster
GRAPHIC 5: Text 80 Zeichen pro Zeile

Die Grafik-Modi 0 bis 4 arbeiten über den Composite-Video-Ausgang, Modus 5 über den RGB-Monitor-Ausgang. Die Grafikauflösung beträgt wie vom C 64 gewohnt 320 x 200 Punkte bei Hochauflösung und 160 x 200 Punkte bei Multicolor. Zusätzlich steht — allerdings mit einigen Tricks auf Maschinensprache-Ebene — noch eine Grafikauflösung von 640 x 200 Punkten über RGB zur Verfügung.

Kann man den C 128 auch mit einem Fernsehgerät als Bildschirm betreiben?

Als einfachste Möglichkeit, um sofort mit dem C 128 arbeiten zu können, bietet sich natürlich ein bereits vorhandenes Schwarzweiß- oder Farbfernsehgerät an. Mit dieser minimalen Grundausstattung können Sie Ihren C 128 bereits betreiben. Dazu brauchen Sie nur das mitgelieferte Antennenkabel in die dafür vorgesehene Buchse an der Rückseite des Computers einstecken und das andere Ende mit dem Antenneneingang Ihres Fernsehers verbinden. Der C 128 »sendet« auf Kanal 36 im UHF-Bereich.

Sie sollten allerdings bedenken, daß Sie mit einem normalen Fernsehgerät den 80-Zeichen-Modus des C 128 (wichtig für CP/M) nicht ausnutzen können.

Bücher zum C 128

Basic 7.0 auf dem Commodore 128

Auch für Einsteiger ist der C 128 aufgrund seines sehr guten Basic ein idealer Computer. Genau an diese Gruppe von Lesern wendet sich dieses Buch aus dem Markt & Technik-Verlag. In den ersten Kapiteln wird man nach und nach mit den Grundzügen der Basic-Programmierung vertraut gemacht. Großen Raum nehmen dabei natürlich die neuen Kommandos des Basic 7.0 ein, wie sie zum Beispiel für strukturiertes Programmieren oder zum Aufspüren und Behandeln von Fehlern vorhanden sind.

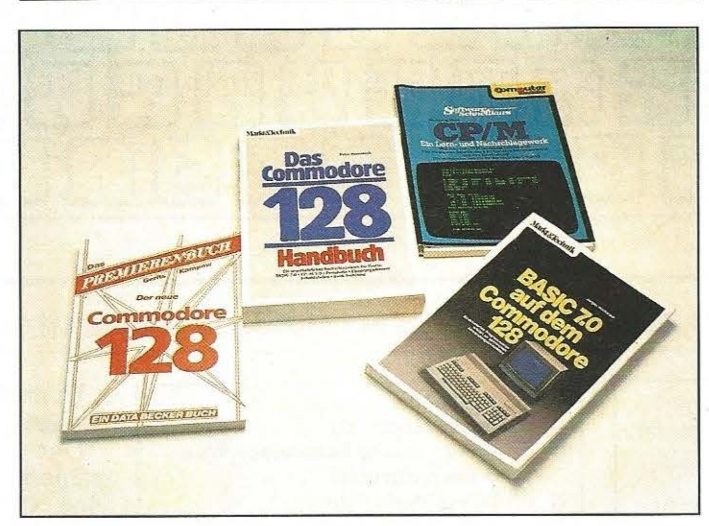
Ein weiteres umfangreiches Kapitel in diesem Buch ist dem Thema Dateiverwaltung gewidmet. Hier kommen besonders die neuen Befehle für die Diskettenstation zum Tragen. Aber auch die Erstellung und Benutzung von sequentiellen und relativen Files sind, dank des neuen Basic 7.0, auch für anfängerorientierte Bücher keine Tabuthemen mehr.

Weitere Schwerpunkte sind die Programmierung von Grafik und Sound. Hier wird speziell auf die hochauflösende Grafik, die neuen Sprite-Befehle und die Sound-Kommandos eingegangen. Wer sich ernsthaft für den C 128 interessiert, der findet in diesen Kapiteln eine gute Grundlage zum Erlernen der vielen, gegenüber dem C 64 neu hinzugekommenen Basic-Befehle.

Viele Beispiellistings sowie Tips und Tricks runden dieses empfehlenswerte Buch ab, das vor allem für den Einsteiger und den Basic-Umsteiger vom C 64 sehr interessant sein dürfte.

(Christoph Sauer/ev)

Info: Jürgen Hückstädt: Basic 7.0 auf dem Commodore 128, Markt und Technik Verlag AG, 250 Seiten, ISBN 3-89090-170-0, 52 Mark



Vier Bücher für Einsteiger und Interessierte

Das Premierenbuch »Der neue C 128«

Kaum ist der C 128 auf dem Markt, findet man bereits die ersten Bücher über diesen Computer in den Läden. Eines dieser Erstlingswerke ist das Premierenbuch von Data Becker »Der neue Commodore 128«. Wie fast alle Bücher, die zur Zeit für den C 128 erhältlich sind, beschäftigt sich auch dieses im wesentlichen mit dem neuen Basic 7.0. Dazu gehört natürlich eine komplette Befehlsübersicht, in der alle neuen (und natürlich auch die alten, vom C 64 bekannten) Befehle und Kommandos beschrieben sind.

Aber auch die interne Organisation des C 128 kommt bereits in diesem Buch nicht zu kurz. Besonders die Aufteilung des Speichers nimmt (verständlicherweise) einen sehr breiten Raum ein. Gerade das Arbeiten mit den verschiedenen Speicherbänken ist ja eine völlige Neuheit, so daß diese Informationen sehr hilfreich sind. Viele ausführliche Grafiken unterstützen die Bemühungen der Autoren, dem Leser die doch nicht ganz einfach zu verstehende Materie nahezubringen.

Den größten Teil dieses Buches nimmt jedoch die Befehlsübersicht des neuen Basic 7.0 ein. Die Darstellung ist ähnlich wie im Handbuch gelöst worden, das heißt auf jeder Seite wird ein Befehl anhand von ein oder zwei kurzen Beispielen erklärt. Diese Aufstellung ist nach Gebieten wie zum Beispiel Grafik, Sound und so weiter sortiert, so daß man sich schnell im umfangreichen Befehlssatz zurechtfindet.

Ein Anhang mit der Belegung der Zeropage schließt dieses Buch ab, das trotz des frühen Erscheinungstermins erstaunlich viele tiefgehende Informationen enthält.

Allerdings muß man anmerken, daß die Befehlsübersicht natürlich auch im erfreulich umfangreichen offiziellen C 128 Handbuch von Commodore zu finden ist (sogar um einiges ausführlicher), so daß dem stolzen Besitzer eines C 128 dieses Buch vermutlich nur wenig Neues sagen wird.

Daher ist das Premierenbuch zum C 128 besonders jenen zu empfehlen, die sich mit dem Gedanken tragen, dieses Gerät anzuschaffen und verständlicherweise nicht die Katze im Sack kaufen wollen.

(Christoph Sauer/ev)

Info: Gerits, Kampow: Das Premierenbuch »Der neue C 128«, Data Becker, 216 Seiten, ISBN 3-89011-062-2, 39 Mark

Das Commodore 128 Handbuch

Nicht nur vom Namen her leicht mit dem Original-Handbuch zu verwechseln ist dieses Buch zum C 128. Das knapp 400 Seiten starke Werk besteht zu 275 Seiten aus einem Anhang, der im wesentlichen aus einer Auflistung aller Befehle des 7.0 Basic besteht. Es gilt hier das gleiche wie für Data Beckers Premierenbuch: Wer sich tatsächlich einen C 128 zulegen will, der bekommt von Commodore ein sehr umfangreiches Handbuch mitgeliefert, das zudem kostenlos ist.

Nun aber zum Buch selbst: Nach Begriffserklärungen im ersten Kapitel geht der Autor zunächst auf die drei unterschiedlichen Betriebsmodi des C 128 ein. Für Hardwarefreaks werden gleichzeitig die dazugehörigen Bausteine auf der Computerplatine kenntlich gemacht und deren Bedeutung erklärt.

Grafik und Sound werden ebenso von »innen heraus« beleuchtet, wie das komplizierte Bankswitching. Gut gelungen ist auch das Kapitel über die Arbeitsweise des eingebauten Maschinensprachenmonitors.

Wer die begehrten Tips und Tricks sucht, der findet in »Das Commodore 128 Handbuch« ein Programm, mit dem man Grafik auch auf dem 80-Zeichen-Bildschirm darstellen kann.

Fazit: das Urteil über dieses Buch fällt gemischt aus. Zum einen enthält es nützliche Informationen für alle diejenigen, die tiefer in ihren C 128 einsteigen möchten. Zum anderen ist da die Befehlsübersicht, die nur für den Nicht-Besitzer interessant ist. Am besten wird man diesem Buch wohl gerecht, wenn man es als allgemeinen Überblick über das C 128-System bezeichnet.

(Christoph Sauer/ev)

Info: Peter Rosenbeck: Das Commodore 128 Handbuch, Markt und Technik Verlag AG, 400 Seiten, ISBN 3-89090-171-9, 52 Mark

CP/M — Ein Lern- und Nachschlagewerk

Hierbei handelt es sich um einen Band aus der Reihe »Software-Schnellkurs« aus dem Markt & Technik-Verlag. Dieses Buch befaßt sich mit CP/M nicht aus der Sicht von Prozessor und Betriebssystem, sondern aus der Sicht des Anwenders. Ein Nachschlagewerk also für diejenigen, die nicht wissen wollen wie CP/M intern arbeitet, sondern die es einfach nur bedienen wollen. Und für diesen Personenkreis ist dieser CP/M-Schnellkurs genau das Richtige.

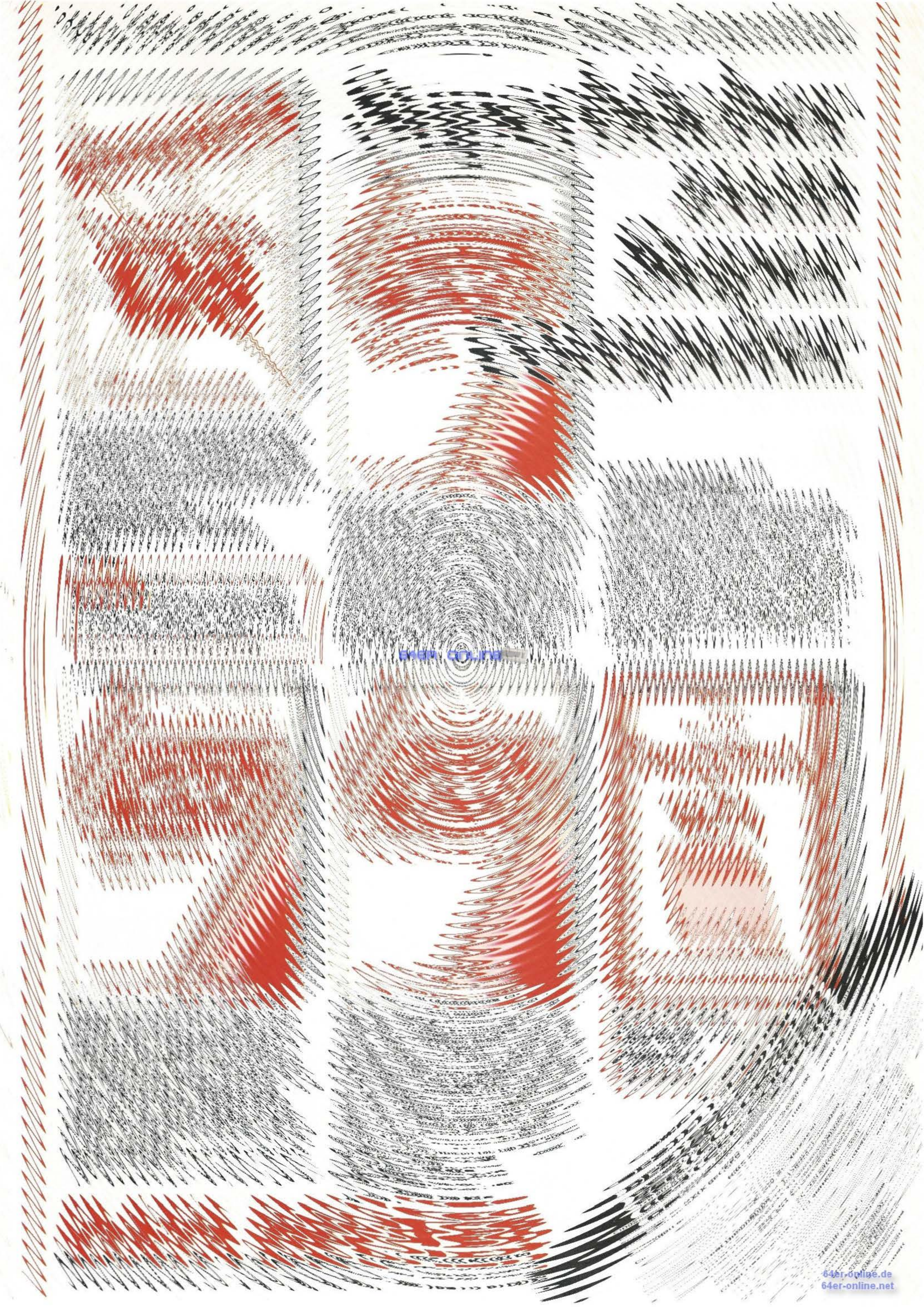
Auf 90 Seiten wird der Anwender auf einfache Weise und mit vielen Beispielen mit den wichtigsten CP/M-Kommandos vertraut gemacht. Gleichzeitig dient das Buch als Nachschlagewerk, falls man einmal etwas vergessen haben sollte. Durch den Ringbuch-Einband kann man das Buch einfach aufgeschlagen neben dem Computer liegen haben, ohne befürchten zu müssen, daß es bei allzu häufiger Benutzung in seine Bestandteile zerfällt.

Leider bezieht sich dieser Schnellkurs noch auf die älteren CP/M-Versionen bis 2.2, was allerdings kein allzugroßer Nachteil ist, da die grundlegenden Kommandos bei allen Versionen die gleichen sind.

Nach der Devise »Lernen durch Benutzen« bekommt man durch dieses Buch schnell Routine bei der Verwendung der einzelnen CP/M-Befehle. Dieser Software-Schnellkurs, den es übrigens auch für spezielle CP/M-Software wie zum Beispiel Wordstar, dBase und Multiplan gibt, ist ein sinnvoller Helfer für den Umgang mit CP/M und gehört neben jeden CP/M-Computer.

(Christoph Sauer/ev)

Info: Wolfgang Maaß: Software-Schnellkurs CP/M, Markt und Technik Verlag AG, ISBN 3-922120-55-5, 37 Mark



Geheimtip: RFI DP 165

Mit dem DP 165 kostet der Einstieg in die Drucker-Spitzenklasse kein Vermögen mehr. Wer keine Angst vor einem unbekannten Namen hat, erhält erstaunlich viel für sein Geld.



Bild 1. Ein echter Profidrucker — Der RFI DP 165

Der DP 165 (Bild 1) ist ein Drucker, der ursprünglich für den professionellen Einsatz an Personal Computern wie dem IBM PC entwickelt wurde. Dementsprechend hoch ist sein Leistungsniveau. Mit mehr als 165 Zeichen pro Sekunde gehört der DP 165 zu den ausgesprochen schnellen Druckern. Unseren Probetext schaffte er in sehr guten 1:48 Minuten und steht in der Rangliste direkt hinter dem Star SR 10, der allerdings fast 600 Mark teurer ist. Trotz der hohen Geschwindigkeit überrascht der DP 165 durch einen angenehm niedrigen Geräuschpegel. Das alleine macht natürlich noch keinen Drucker der Spitzenklasse aus. Wie aus der Tabelle zu sehen ist, bietet der DP 165 eine enorme Vielfalt an Funktionen und Schriftarten. Bei der Syntax der Steuerbefehle glänzt der DP 165 durch vollständige Gleichheit mit dem Epson FX-80, dessen Befehle schon fast als Standard gelten können. Der Unterschied zum FX-80 liegt in der NLQ-Schrift (Near Letter Quality = Schöndruck), die sich sogar mit proportionalen Zeichenabständen drucken läßt (Bild 2).

Epson-Steuerbefehle

Die NLQ-Schrift ist auf drei verschiedene Arten einzustellen. Einmal durch einen Steuerbefehl, durch eine separate Taste auf dem Bedienfeld und durch einen der gut erreichbaren DIL-Schalter auf der Gehäuserückseite. Die Qualität dieser Schrift kann sich sehen lassen (Bild 3), zumal der DP 165 in der NLQ-Schrift immer noch schneller (und leiser) druckt als die meisten Typenraddrucker. Normalerweise ist das Aussehen der NLQ-Schrift durch die im ROM des Druckers abgelegten Werte bestimmt (17 x 17 Matrix). Erweitert man das interne RAM von 2 KByte auf 8 KByte, wird es sogar möglich, eine eigene NLQ-Schrift zu entwerfen. Ohne die RAM-Erweiterung bleibt der selbstdefinierte Zeichensatz auf die, schon vom FX-80 her bekannte, 9 x 9 Zeichenmatrix beschränkt. Eine andere Methode, selbstdefinierte Zeichen und Grafiken auszudrucken, besteht in der Verwendung der umfangreichen Grafik-Fähigkeiten des DP 165. Alle Hardcopy-Routinen, die für die Epson-Drucker programmiert wurden, funktionieren, ohne jede Änderung, auch zusammen mit dem DP 165. Besitzt man das optionale Commodore-Interface (235 Mark), kommen sogar noch die Grafik-Routinen für dem MPS 801-Drucker hinzu (Simons Basic, Supergrafik).

Der praktische Wert eines Druckers hängt allerdings nicht nur von seinen Fähigkeiten, sondern auch vom Bedienungskomfort ab. Dazu gehört ein einfach zu bedienender Papiereinzug, ein Papiertransport, der ein direktes Abreißen des Papiers über dem Druckkopf ermöglicht und eine Druckersteuerung über Funktionstasten. Alle diese Bedingungen erfüllt der DP 165. Das Papier, das entweder Einzel- oder Endlospapier mit Randlochung sein kann, wird ohne großen

Der RFI DP 165
Der DP 165 ist ein professioneller Drucker
Er druckt breit und schmal
Das ist die NLQ-Schift im Proportional-Modus
Alle anderen Funktionen sind wie beim FX 80

Bild 2. Der DP 165: Near Letter Quality-Schrift und alle FX 80-Funktionen

Name des Druckers	: RFI DP 165
Schriftarten	: Pica schnell, Pica schön, Elite, komprimiert, Doppeldruck, Fettschrift, Breitschrift
Geschwindigkeit	: angegeben 165 Z/s, gemessen 169 Z/s, Probetext: 1:48
NLQ-Schrift	: ja, 17 x 17 — NLQ-Geschw.: 41 Z/s
Unterstreichen	: ja — Proportional: ja
Zeichenmatrix	: 9 x 9 — Zeichenvorrat: 96 ASCII, 96 Ital., 8 Int.
Papierarten	: Einzel/Endlos — Durchschläge: 3
Spaltenbreite v-b	: 0 bis 136 — Papierbreite: 10,1 bis 25,4 Zentimeter
Selbsttest	: ja, Hexdump: ja — Automat. Einzelbl.-Einz.: nein
Schnittstellen	: Centronics (Standard), RS232, IEEE 488, CBM m. ext. Interf.
Pufferspeicher	: 2 KByte — Ladbarer Zeichens.: ja, 96 Zeichen
Rückwärtstransp.	: ja — Hor. Tab.: ja — Ver. Tab.: ja
Grafikmodi	: 480, 960, 960 Punkte pro Zeile
Sonderfunktionen	: variabler Zeilenabstand, Setzen rechter, - linker Rand, Hochstellen, Tiefstellen, unidirektionaler Druck, rechter und linker Rand mit Tasten einstellbar
Funktionstasten	: On-Line, LF, FF
Ausstattung	: Farbband, deutsche u. englische Handbücher, Papierführung

Tabelle. Die wichtigsten Daten des DP 165

Aa Bb

Bild 3. Erst bei fünffacher Vergrößerung treten Einzelpunkte hervor

Aufwand von hinten in den Drucker eingelegt und durch zwei Stachelwalzen transportiert. Die Stachelwalzen sind in ihrem Abstand beliebig verstellbar. Auch das Einlegen des Farbbandes ist nicht schwieriger als bei einem FX-80, zumal dessen Farbbänder problemlos passen. Die vier Bedienungstasten des DP 165 haben neben den Standardfunktionen wie Zeilen- und Seitenvorschub noch eine weitere Bedeutung erhalten. Mit ihnen lassen sich auf einfache Weise die linken und rechten Ränder des Schreibfeldes einstellen.

Problemloser Anschluß

Um den DP 165 an den C 64 oder C 128 anzuschließen, gibt es mehrere Möglichkeiten. Normalerweise ist der DP 165 mit einer Centronics-Schnittstelle ausgestattet. Wer keinen Wert auf die Commodore-eigenen Steuerzeichen legt, kann sich mit einem einfachen User-Port-Kabel (zirka 50 Mark) behelfen. Daneben wird vom Hersteller ein externes Interface angeboten, das für eine vollständige Anpassung an die Belange der Commodore-Computer sorgt. Der DP 165 wird dann einfach am seriellen Bus des Computers, wie ein Floppy-Laufwerk angeschlossen. Die dritte Möglichkeit besteht darin, ein serielles RS232-Interface einzubauen und den Drucker wie einen Akustikkoppler anzusprechen.

Der DP 165 ist ohne Zweifel ein Drucker der Spitzenklasse. Mit einem Preis von 1498 Mark hält er sich, gemessen an seinen Fähigkeiten (Tabelle), in erfreulich bescheidenen Grenzen. Seine vollständige Übereinstimmung mit den Funktionen der Epson FX-80-Drucker, ergänzt durch die NLQ-Schrift und den niedrigen Geräuschpegel, zeichnen den DP 165 aus. Mit diesen Leistungen gehört der DP 165 in eine Klasse mit dem Star SR-10 und dem Centronics Horizon. Sein um zirka 600 Mark unter dem dieser Konkurrenten liegender Preis, machen ihn zusätzlich interessant. Wer sich etwas abseits von den Pfaden der größeren Druckerhersteller umschaut, kann also durchaus so manche Mark sparen. (aw)

Info: RFI Elektronik GmbH, Mittelstr. 12a, 4050 Mönchengladbach 2, Telefon 021 66/4 80 77, Preis: 1498 Mark



Bild 1. Der CP-80X, ein Schnittstellen-Meister

Melchers CP-80X — wie hätten Sie's denn gern?

Mit seinen drei eingebauten Schnittstellen kennt der CP-80X keine Anschlußprobleme. Lesen Sie, warum der CP-80X so gut zum C 64 und C 128 paßt.

Den CP-80X (Bild 1) könnte man beinahe als Chamäleon bezeichnen, denn er versteht es, sich ganz an die Schnittstellen der verschiedensten Computer anzupassen. Die Konstrukteure des CP-80X haben ihn gleich mit den drei der gängigsten Normen (außer RS232) ausgerüstet: einer Centronics parallel, einer seriellen IEC- (wie sie der C 64/C 128 hat) und einer parallelen IEC-Schnittstelle. Für uns war natürlich seine serielle IEC-Schnittstelle am wichtigsten. Der Anschluß des CP-80X ist sehr einfach, er wird mit einem Kabel direkt an den Computer, beziehungsweise das Floppy-Laufwerk, angeschlossen. Danach ist der CP-80X sofort einsatzbereit. Trotzdem sollte man nicht sofort mit dem Drucken beginnen. Es lohnt sich, zunächst das ausführliche, in Deutsch geschriebene, Handbuch durchzulesen. Im Unterschied zu den meisten Drucker-Handbüchern sind die dort abgedruckten Beispieleprogramme im Basic des C 64 (das im Basic des C 128 enthalten ist), geschrieben. Die Einarbeitung in die umfangreichen Fähigkeiten des CP-80X sollten darum auch einem Anfänger leichtfallen. Die Steuerbefehle des CP-80X stellen eine gelungene Kombination aus den Befehlen der Commodore MPS-Reihe und denen eines Epson RX-80 dar. Dadurch wird erreicht, daß die meisten Programme, die für die MPS-Reihe geschrieben wurden, mit wenigen Änderun-

Name des Druckers	: CP-80X		
Schriftarten	: Fettdruck, Doppeldruck, komprimierte Schrift, vergrößerte Schrift		
Geschwindigkeit	: angegeben: 80 Z/s, gemessen: 83 Z/s, Probetext: 3:25		
NLQ-Schrift	: nein	NLQ-Geschw.: entfällt	
Unterstreichen	: ja	Proportional.: nein	
Zeichenmatrix	: 8 x 8	Zeichenvorrat: CBM +ä,ö,ü,ß,Ä,Ö,Ü	
Papierarten	: Einzel-/Endlos	Durchschläge: 3	
Spaltenbreite v-b	: 0 bis 143	Papierbreite: 101,6 bis 254 mm	
Selbsttest	: ja, Hexdump: nein	Automatischer Einzelbl.Einz.: nein	
Schnittstellen	: Centronics parallell, IEC seriell, IEC parallel,alle eingebaut		
Pufferspeicher	: nein	Ladbarer Zeichens.: nein	
Rückwärtstransp.	: nein	Hor. Tab.: ja	Ver. Tab.: ja
Grafikmodi	: 480 und 960 Punkte pro Zeile		
Sonderfunktionen	: Hoch- und Tiefstellen, unidirektionales Drucken		
Funktionstasten	: On Line, FF, LF		
Ausstattung	: Drucker, Handbuch, Farbband, Anschlußkabel, Papierhalter		

Tabelle. Die wichtigsten Daten des CP-80X

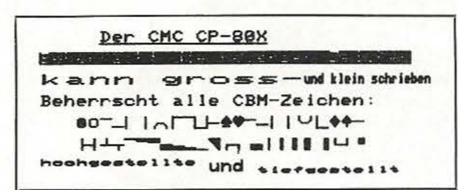


Bild 2. Die vielseitigen Schriftarten des CP-80X verkleinert

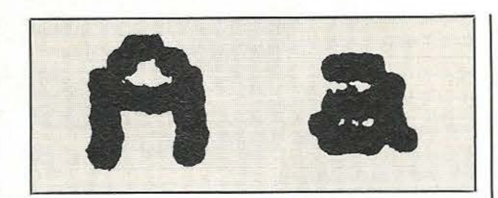


Bild 3. Der fünffach vergrößerte Buchstabe A der Normalschrift

gen funktionieren. Es ist erfreulich, daß eine Druckersteuerung mit den bekannten ESC-Befehlen trotzdem erhalten wurde. Unser Probeausdruck (Bild 2) zeigt, daß alle Funktionen des Druckers ohne viel Aufwand programmiert werden können. Die reverse Schrift wurde beispielsweise durch die Verwendung der CTRL- und RVS ON-Taste innerhalb der Anführungsstriche erzeugt. Die Druckersteuerung erfolgt somit bei einigen Befehlen analog zur Cursorsteuerung auf dem Bildschirm. Drucker kann man aber nicht nur sehen, sondern auch hören. Der CP-80X gehört zu den Druckern, die man, ohne Angst vor den Nachbarn, auch in der Etagenwohnung betreiben kann. Überhören kann man ihn — wie jeden Nadel-Matrixdrucker — allerdings nicht. Erfreulich war auch das Ergebnis unseres Geschwindigkeitstests. Im Handbuch wird die Geschwindigkeit des CP-80X mit 80 Zeichen pro Sekunde angegeben, unseren Messungen nach schaffte das Testgerät aber gute drei Zeichen mehr. Dieser Vorteil macht sich natürlich erst bei längeren Texten, wie unserem Probetext, den er in der Zeit von 3:25 ausdruckte, bemerkbar. Zu den Fähigkeiten eines guten Matrixdruckers wie dem CP-80X gehört neben der Darstellung von Texten auch die Grafik. Der CP-80X wird diesem Anspruch in jeder Hinsicht gerecht, denn seine Grafikauflösung reicht von 480 bis zu 960 Punkten pro Zeile. Die Verwendung einiger Hardcopy-Routinen, wie beispielsweise der COPY-Befehl von Simons Basic, funktionieren allerdings nicht auf Anhieb. Der Hersteller legt dem CP-80X für diesen Zweck ein eigenes kleines Zusatz-Handbuch mit Hilfsprogrammen bei. Eine Übersicht aller weiteren technischen Daten finden Sie in der Tabelle.

Schneller als angegeben

Wenden wir uns nochmals den Schnittstellen des CP-80X zu. Mit den drei eingebauten und einer nachrüstbaren RS232-Schnittstelle ist der CP-80X mehr als üppig ausgestattet. Das Umschalten zwischen den einzelnen Schnittstellen erfolgt über eine Reihe von DIL-Schaltern. Je nachdem, wie diese Schalter eingestellt sind, belegt der CP-80X die ausschließlich vorhandene Centronics-Buchse mit einer anderen Norm. Das bedeutet, daß er ohne zusätzliche Erweiterungen sowohl am C 64/C 128, als auch an verschiedene andere Computer direkt anschließbar ist. Einzig der für die DIL-Schalter gewählte Platz war Anlaß zur Kritik, denn sie befinden sich innerhalb des Gehäuses an unzugänglicher Stelle.

Der CP-80X ist nicht nur wegen seiner Schnittstellen, sondern auch durch seine Funktionsvielfalt ein universeller Drucker. Jeder, der Texte bearbeiten und korrekte Listings mit allen Cursor-und Grafikzeichen ausdrucken möchte, findet einen fast vollständigen CBM-Zeichensatz vor. In einigen wesentlichen Details unterscheidet sich dieser Zeichensatz allerdings von denen der Commodore-Drucker — er wurde um die deutschen Umlaute ergänzt. Leider entsprechen die CHR$-Werte für die Umlaute bei den bisher verkauften Geräten noch nicht der normalen ASCII-Tabelle. Nach Aussagen des Herstellers wird ein neues Zeichensatz-ROM, bei dem eine Standard-ASCII-Tabelle zugrunde gelegt wird, gerade entwickelt. Mit dieser Änderung wäre der CP-80X dann auch für den DIN-Modus des C 128 geeignet.

Preiswert und flexibel

Der CP-80X ist mit einem Preis von 889 Mark als grafikfähiger Drucker wohl hauptsächlich durch seine Vielseitigkeit interessant. Obwohl sein Schriftbild (Bild 3) durchaus als gut bezeichnet werden kann, fehlt doch die, heute schon fast zum Standard gewordene, NLQ-Schrift. Diesen Nachteil wiegt der CP-80X allerdings durch seine umfangreichen Befehle und die gut gelungene Anpassung an den C 64 wieder auf.

(aw)

Info: Melchers & Co., 2800 Bremen 1, Schlachte 39/40, Tel.: 0421/176989

Epson GX-80. Einer für alle

Der Epson GX-80 präsentiert sich mit einem völlig neuen Konzept: den Steckmodul-Schnittstellen. Wir haben Vor- und Nachteile verglichen.

Ganz anders, als die bisherigen Druckergenerationen von Epson, besitzt der GX-80 (Bild 1) keine Schnittstelle im herkömmlichen Sinne. An Stelle einer Anschlußbuchse hat der GX-80 einen Modulschacht. Durch dieses Konzept soll eine möglichst perfekte Anpassung an die verschiedensten Computer erreicht werden. Das in jedem Fall notwendige und im Preis enthaltene Interface wird einfach in den Modulschacht eingesteckt und verleiht dem GX-80 alle Fähigkeiten, die er braucht, um an dem betreffenden Computertyp zu arbeiten. Wir haben den GX-80 mit Commodore-Modul getestet. In dieser Version besitzt der GX-80 eine Schnittstelle, die am seriellen Bus des C 64 angeschlossen wird. Die Anpassung an die Besonderheiten des C 64, wie Grafik- und Steuerzeichen, ist damit total. Es ist beinahe unmöglich festzustellen, daß der GX-80 nicht von Commodore selbst stammt. Alle Programme, die für den MPS 802, beziehungsweise MPS 801 geschrieben wurden, sind ohne große Änderungen auf dem GX-80 funktionsfähig. Auch in seinem äußeren Erscheinungsbild paßt der GX-80 sehr gut zu den Commodore-Computern, insbesonders zum C 128. Die Reihe der Neuerungen hört aber nicht beim Interface-Konzept auf. Der GX-80 ist der einzige Epson-Drucker dieser Preisklasse, der über die sogenannte NLQ-Schrift (Near Letter Quality = Schönschrift) verfügt. Die NLQ-Buchstaben des GX-80 sind im Gegensatz zum Star SR-10 etwas runder und nicht ganz so scharf geschnitten (Bild 2). Aus normaler Leseentfernung fallen die Abstände zwischen den einzelnen Matrixpunkten allerdings kaum noch auf. Außerdem besitzt der GX-80 einen einfachen Grafikmodus mit 480 Punkten pro Zeile. Damit erreicht der GX-80 allerdings nicht die Lei-

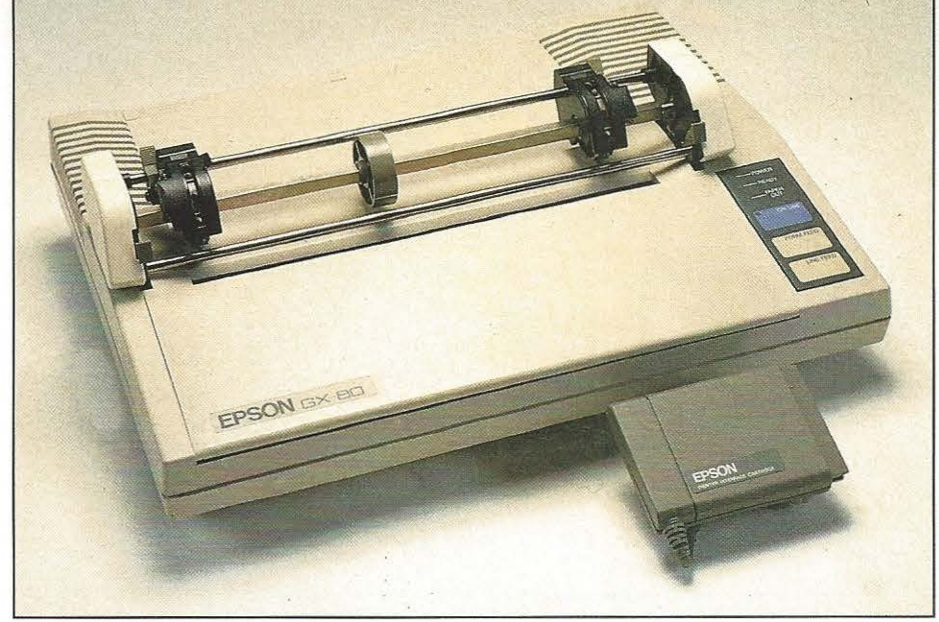


Bild 1. Der GX-80, ein Drucker mit »versteckten« Fähigkeiten

Aa Bb

Bild 2. Die fünffach vergrößerte NLQ-Schrift zeigt deutliche Einzelpunkte

stungen des seit langem bekannten RX-80. Dafür beherrscht der GX-80 alle Commodore-spezifischen Zeichensätze. Der korrekte Ausdruck von Listings ist somit sichergestellt. Anders sieht es bei der Eignung zur Textverarbeitung aus. Trotz der Druckgeschwindigkeit von 100 Zeichen pro Sekunde und der NLQ-Schrift eignet sich der GX-80 für diese Anwendung weniger. Es war uns nicht möglich, den deutschen Zeichensatz mit und ohne Textverarbeitungsprogramm zu realisieren. Wahrscheinlich wurde zugunsten einer vollkommenen Anpassung an die Commodore-Computer darauf verzichtet. Auch die Druckgeschwindigkeit entspricht nicht den Erwartungen. Mit einer Zeit von 3:30 für unseren Probetext konnte der GX-80 nicht ganz befriedigen.

Der Heimarbeiter

Der GX-80 ist eindeutig für die Heimanwendung konzipiert worden. Dafür sprechen nicht nur die einfache Handhabung, sondern auch die neue Konstruktion des Papierantriebs. In der Grundausstattung ist der GX-80 mit einem Friktionsantrieb ausgestattet. Damit lassen sich nur Einzelblätter und Rollenpapier verarbeiten. Gegen einen Aufpreis von 82 Mark ist ein Traktorantrieb, dessen Montage problemlos ist (einfach aufstecken), erhältlich. Zu dieser Investition ist in jedem Fall zu raten, denn Endlospapier mit Randlochung ist immer noch die preiswerteste Art, größere Papiermengen zu bedrucken. Ein, gelegentlich recht sinnvoller, Einzelblatteinzug wird für zusätzliche 219 Mark angeboten.

Weit gewichtiger als das Fehlen eines Traktorantriebs (in der Grundausstattung) ist die Begrenzung der programmierbaren Funktionen (Bild 3) auf die der Commodore-Drucker. Alle Variationen der Schrift (Fett, NLQ, komprimiert) sind nur mit einem ebenfalls neuen Verfahren einzustellen. In praxi sieht das folgendermaßen aus: man betätigt die »On Line«- und »Form Feed«-Taste gleichzeitig und befindet sich im »Select«-Modus. Mit der »On Line«-Taste wird die gewünschte Schriftart (Tabelle) ausgewählt und mit der »Form Feed«-Taste eingespeichert. Dieses Verfahren mag für den Anwender, der den Drucker nicht selbst programmieren möchte, sicherlich recht praktisch sein. Jeder, der seinen Drucker mit allen Funktionen, auch programmgesteuert, flexibel einsetzen möchte, wird die »Handprogrammierung« schon bald ablehnen.

```
Der Epson GX-80
Die NLQ-Schrift ist neu bei Epson
Dies ist der Fettdruck
Auch der Doppeldruch laest sich einstellen
Im komprimierten Druck passen 136 Zeichen in eine Zeile
Programmierte Breitschrift
Revers gehoert dazu
Grafikzeichen:
```

Bild 3. Die Schriften des GX-80 sind nicht alle programmierbar

Name des Druckers	: Epson GX-80		
Schriftarten	: Breit, revers, komprimiert, fett, doppelt		
Geschwindigkeit	: angegeben: 100 Z/s, gemessen: 98 Z/s		
NLQ-Schrift	: ja	NLQ-Geschw.: 20 Z/s	
Unterstreichen	: nein	Proportional.: nein	
Zeichenmatrix	: 8 x 8	Zeichenvorrat: CBM	
Papierarten	: Einzel/Endlos	Durchschläge: 2	
Spaltenbreite v-b	: 0 bis 136	Papierbreite: 101 bis 205 mm	
Selbsttest	: ja, Hexdump: ja	Automatischer Einzelbl. Einz.: nein (option)	
Schnittstellen	: IEC seriell, verschiedene Module		
Pufferspeicher	: nein	Ladbarer Zeichens.: nein	
Rückwärtstransp.	: nein	Hor. Tab.: nein	Ver. Tab.: nein
Grafikmodi	: 480 Punkte/Zeile		
Sonderfunktionen	: Einstellen der Schriftarten mit den Funktionstasten		
Funktionstasten	: On Line, FF, LF		
Ausstattung	: Drucker, Handbuch, Papierseparator, Interface, Farbband		

Tabelle. Die wichtigsten Daten des GX-80

NLQ-Schrift zum Niedrigpreis

Mit dem Preis von 998 Mark (plus 82 Mark für den Traktor) ist der GX-80 einer der preiswertesten Drucker mit NLQ-Fähigkeit auf dem Markt. Sein gutes Aussehen, der niedrige Geräuschpegel und die vollkommene Anpassung an den C 64 sind Argumente, die für diesen Drucker sprechen. Leider wurde bei der Realisierung der Commodore-Anpassung einiges der prinzipiell vorhandenen Fähigkeiten verschenkt. Ein Drucker, ohne die programmgesteuerte Funktionswahl und deutsche Umlaute, kann den heutigen Ansprüchen, auch im Heimbereich, nicht mehr gerecht werden. Es bleibt zu hoffen, daß diese feinen, aber wesentlichen Punkte noch korrigiert werden. Erst dann wird der GX-80 zum universell verwendbaren Drucker, für fast jeden Zweck, zu einem interessanten Preis. (aw)

Info: Epson Deutschland GmbH, Am Seestern 24, 4000 Düsseldorf 11, Tel. 0211/59520



Die Beschleunigungssysteme für die 1541 werden immer besser. Zwei Systeme werden hier wohl das Rennen um die Gunst der Käufer machen: SpeedDos plus und Prologic Dos. Wir haben sie getestet.

Beschleuniger für die 1541 gibt es inzwischen viele: Von der reinen Software-Lösung (Hypra-Load) bis zur Installation eines parallelen Busses. Am schnellsten und sichersten sind die letztgenannten Systeme, leider aber auch am teuersten.

Die zwei derzeit wohl besten Hardware-Erweiterungen haben wir unter die Lupe genommen. Sie zeichnen sich beide durch Geschwindigkeit, Komfort und Zusatzfunktionen aus.

SpeedDos plus — klein, aber oho!

Da wäre erst einmal SpeedDos plus, die verbesserte Version von SpeedDos. Geliefert werden zwei neue Betriebssysteme und ein Kabel zum Preis von 269 Mark. Interessant ist SpeedDos hauptsächlich deswegen, weil es inzwischen schon einige Programme gibt, die die SpeedDos-Hardware ausnutzen, so zum Beispiel das Kopierprogramm »FCopy III«, (vorgestellt in 64'er 8/85), das Schnelladesystem »FLoad 3.0« (siehe Kasten) und das Kopierprogramm »Copy +« (siehe unter Aktuell).

Wie gesagt, ist SpeedDos plus eine leicht verbesserte Version des bekannten SpeedDos. Da wir SpeedDos aber schon in der Ausgabe 4/85, Seite 22, ausführlich getestet haben, wollen wir hier nur einmal einen groben Überblick geben und die Unterschiede zu SpeedDos selbst aufzeigen.

Die SpeedDos-Hardware ist relativ einfach aufgebaut. Die parallele Busübertragung läuft über ein zehnpoliges Flachbandkabel. Dieses wird am User-Port des C 64 angeschlossen. Auf der Laufwerksplatine muß zum Anschluß ein Baustein (6522) gesockelt werden.

Ins Betriebssystem des Computers wurden integriert: eine Basic-Erweiterung ähnlich dem DOS-Wedge auf der Commodore-Demo-Diskette, eine Centronics-Schnittstelle (ebenfalls am User-Port), eine Funktionstastenbelegung, ein Mini-Monitor sowie Verbesserungen an der LIST- und der RESET-Routine.

Auf der Laufwerksseite finden sich an Verbesserungen: schnelles Formatieren mit vollem Verify (23 Sekunden), Löschschutz für Files, Zentrieren der Diskette beim Einlegen sowie schnellere Tonkopfbewegungen.

Für SpeedDos plus wurden zusätzliche Veränderungen im Betriebssystem des Computers vorgenommen. Der parallele Bus und das Laufwerks-Betriebssystem ist identisch zu SpeedDos. Folgende Veränderungen hat SpeedDos plus zu bieten: eine OLD-Funktion, Hardcopies vom Textbildschirm, Rechnungen in Binär, Oktal und Hexadezimal von Basic aus, sowie einige kleine Editor-Verbesserungen.

Die Verträglichkeit der Erweiterung zu professioneller Software ist sehr hoch: Sollte ein Programm mit den schnellen Lade-Routinen von SpeedDos nicht laufen, kann man auf langsamere, aber verträglichere Routinen durch Angabe der Sekundäradresse Zwei beim LOAD-Befehl umschalten. Im Notfall zieht man einfach den User-Port-Stecker ab, und schon wird wieder ganz normal (und langsam) geladen. Wir konnten während der Testzeit kein Programm auftreiben, das sich nicht zur Zusammenarbeit mit SpeedDos plus bewegen ließ. Fast hätten wir's vergessen: Natürlich macht SpeedDos plus Ihre Floppy schneller. Wieviel schneller, das können Sie aus der Tabelle ersehen, in der ein paar typische Zeiten festgehalten sind.

Prologic Dos — Das Geschwindigkeitswunder

Der zweite Beschleuniger heißt Prologic Dos und ist eine kleine Sensation: 25mal schneller laden wurde versprochen und das mit normal gespeicherten Programmen. Um dies zu erreichen, mußte allerdings ein etwas größerer Aufwand bei der Laufwerks-Hardware getrieben werden. Trotz der komplizierten Erweiterungsplatine im Laufwerk soll Prologic Dos nur knapp 300 Mark kosten. Durch die eingesteckte Zusatzplatine wird beispielsweise erreicht, daß ein kompletter Track im Laufwerks-RAM zwischengespeichert werden kann. Die Daten, die sich auf einem Track befinden, können in einer Umdrehung der Diskette gelesen werden. Dann sind vorerst keine Zugriffe auf die Diskette mehr nötig, erst wieder bei einem Trackwechsel. In der Zwischenzeit können die angeforderten Daten sofort aus dem RAM an den Computer weitergegeben werden. Ähnlich funktioniert das Schreiben. Hier wird der Track im RAM zusammengestellt und erst wenn er fertig ist, auf die Diskette geschrieben. Welche fantastischen Lade- und Speichergeschwindigkeiten damit möglich sind, können Sie unserer Tabelle entnehmen.

Wie gesagt, ist die Prologic Dos-Hardware etwas umfangreicher als die von SpeedDos. Eine große Zusatzplatine für das Laufwerk und eine weitere für den Expansionport des Computers werden geliefert. Der Expansionport geht aber nicht verloren, da er durchgeführt ist. Verbunden sind die beiden mit einem Flachbandkabel, welches gleichzeitig den parallelen Bus darstellt. Das Betriebssystem des Computers muß nicht ausgewechselt werden, da sich das neue Betriebssystem ebenfalls auf der Platine im Expansionport befindet.

Aber neben der Beschleunigung bietet Prologic Dos noch einiges an Bedienungskomfort. So hat man jetzt die Möglichkeit, Disketten vierzigspurig zu formatieren, zu beschreiben und zu lesen. Dadurch ergibt sich ein Speicherplatzgewinn von 85 Blöcken oder mehr als 20 KByte pro Diskette. Diese Option ist soft- und hardwaremäßig schaltbar. Außerdem kann man von Basic aus an beliebige Speicheradressen laden. Es läßt sich auch das RAM unter dem Basic-ROM ($A000-$BFFF) auf Diskette speichern.

Auf der Computerseite findet man bei Prologic Dos die fast schon üblichen Optionen: belegte Funktionstasten, eingebautes DOS, modifizierte LIST- und Reset-Routinen, Tastatur-Repeat, Textschirm-Hardcopy, sowie eine Centronics-Schnittstelle, die allerdings nicht den User-Port benutzt, sondern sich auf der Platine im Expansionport befindet. Diese Schnittstelle ist aber vollkommen zu User-Port-Schnittstellen kompatibel, das heißt, daß die üblichen Kabel verwendet werden können. Programme, die eingebaute User-Port-Centronics-Schnittstellen haben, können natürlich weiter genutzt werden, dann muß man sein Druckerkabel auf den User-Port aufstecken. Übrigens sind, und das wird Besitzer von Akustikkopplern freuen, die RS232-Routinen bei Prologic Dos vollständig erhalten geblieben.

Auch bei der 1541 selber hat sich einiges getan. So hat das Laufwerk ein neues Kommando bekommen: »X«. Mit diesem Kommando werden eine Reihe von Unterfunktionen gesteuert: Schreibschutz für Einzelfiles installieren und aufheben, Zusatz-RAM in Floppy ein- und ausschalten, schnelle Übertragung ein- und ausschalten, Verify nach Schreibzugriffen ein- und ausschalten sowie die Umschaltung 40/35 Tracks.

Ein Fehler der 1541 wurde übrigens mit einem Trick übergangen: Die Replace-Option beim SAVE-Befehl (»@«) funktionierte bisher nicht vernünftig. Bei Prologic Dos wird bei einem Replace einfach das alte File per Scratch gelöscht und dann das neue normal gespeichert.

Prologic Dos kann ebenfalls eine sehr hohe Verträglichkeit zu fertiger Software bescheinigt werden. Auch hier gilt: sollte ein Programm mal nicht laufen, genügt meistens ein Abschalten der schnellen Busroutinen, um zum gewünschten Ergebnis zu kommen. Lobenswert gegenüber SpeedDos ist, daß dies auch softwaremäßig geschehen kann. Und für ganz hartnäckige Fälle kann man Prologic Dos per DIP-Schalter komplett abschalten, so daß man wieder einen »jungfräulichen« Commodore 64 vor sich stehen hat.

Was Prologic Dos aber gegenüber SpeedDos weit hinaushebt, ist sein Preis-/Leistungs-Verhältnis. Trotz des Mehraufwands aufgrund der beiden umfangreichen Platinen soll es mit 298 Mark nicht viel mehr kosten als SpeedDos plus (269 Mark).

Spätestens mit Prologic Dos hat die zweite Generation der Floppy-Beschleuniger begonnen. SpeedDos hat zwar die Vorteile eines hohen Verbreitungsgrades und fertiger Software. Trotzdem ist Prologic Dos mit seinen Leistungen und seinem Preis mehr als nur Konkurrent im Rennen um den besten Floppy-Beschleuniger. Man darf gespannt sein, was die nächsten Monate uns noch bringen werden. (aw/bs)

Info: SpeedDos plus: Elektronik-Service Christoph Dichte, Fährstr. 33, 2212 Brunsbüttel
Prologic Dos: Jann Datentechnik, Glimmerweg 22, 1000 Berlin 47.

	1541 normal	SpeedDos plus	Prologic Dos
Laden 202 Blocks	2 min 11 sec	25 sec 16 sec[1]	5 sec
Speichern 202 Blocks	2 min 25 sec	1 min 41 sec	10 sec
Formatieren	88 sec	23 sec	21 sec
Block- Read[2] **683 Blocks**	4 min 40 sec	2 min 35 sec	2 min 01 sec

Bemerkungen: [1] — wenn mit SpeedDos plus gespeichert wurde, [2] — Basic-Programm, das jeweils die ersten beiden Bytes eines Blocks liest

Tabelle. Zeitvergleich zwischen der »normalen« 1541, SpeedDos plus und Prologic Dos.

30mal schneller mit SpeedDos

Wer SpeedDos schon besitzt, dem ist mit FLoad 3.0 eine billige Alternative zu Prologic Dos gegeben: FLoad 3.0 ermöglicht bis zu 30mal schnelleres Laden von Programmen von der 1541. Das System hat allerdings einen Nachteil: Damit Programme so schnell geladen werden können, müssen sie auf ein eigenes Format umkopiert werden. Dies bereitet bei kopiergeschützter Software verständlicherweise Probleme. Ein Testfile von 202 Blöcken wird mit FLoad 3.0 in weniger als sechs Sekunden geladen. Davon entfallen 2,5 Sekunden auf die Installation der neuen Busroutinen. Es ist nicht unbedingt notwendig, SpeedDos zu besitzen, um dieses Programm zu nutzen. Ein paralleles Kabel allein reicht. Dieses Kabel wird demnächst ebenfalls beim Vertrieb von FLoad 3.0 erhältlich sein. FLoad 3.0 soll unter fünfzig Mark kosten. (bs)

Info: Frank Thomas, Neckarstr. 34, 6057 Dietzenbach 2

Professionelle Floppy-

Das 1541-Laufwerk ist neben dem 1571 das einzige von Commodore hergestellte Diskettenlaufwerk, das an den C 64 direkt anschließbar ist. Gleichzeitig ist es ab er auch das langsamste. Warum bei der Entstehung des C 64 ausgerechnet die 1541 (damals 1540) favorisiert wurde, obwohl Commodore auch andere, wesentlich leistungsfähigere Laufwerke herstellt, ist wahrscheinlich ein Relikt aus VC 20-Zeiten. Der Preis ist, zumindest heute, kaum noch ein Grund auf ein größeres Laufwerk zu verzichten. Die SFD 1001, ein sehr schnelles Laufwerk mit einem MegaByte Speicherkapazität, wird inzwischen schon für 698 Mark im Handel angeboten und ist damit nicht wesentlich teurer als die 1541. Auch die 8250 (alte Version), ein Doppellaufwerk mit zwei Megabyte, kostet heute weniger als 2 000 Mark. Diese Laufwerke lassen sich leider nicht direkt am C 64 anschließen, da sie einen parallelen IEEE-488-Bus voraussetzen. Dieser Bus arbeitet wesentlich schneller als der serielle IEC-Bus des C 64. Natürlich können serielle Schnittstellen auch recht flott sein (über 19200 Bit pro Sekunde), aber beim seriellen IEC-Bus kommt ein weiteres Manko hinzu: Das gesamte Bus-Timing wird durch Software gesteuert, die umständlich und langsam arbeitet. Deshalb gibt es dort auch so viele Ansatzpunkte für Beschleunigungsroutinen, wie zum Beispiel das Programm »Hypra-Load«.

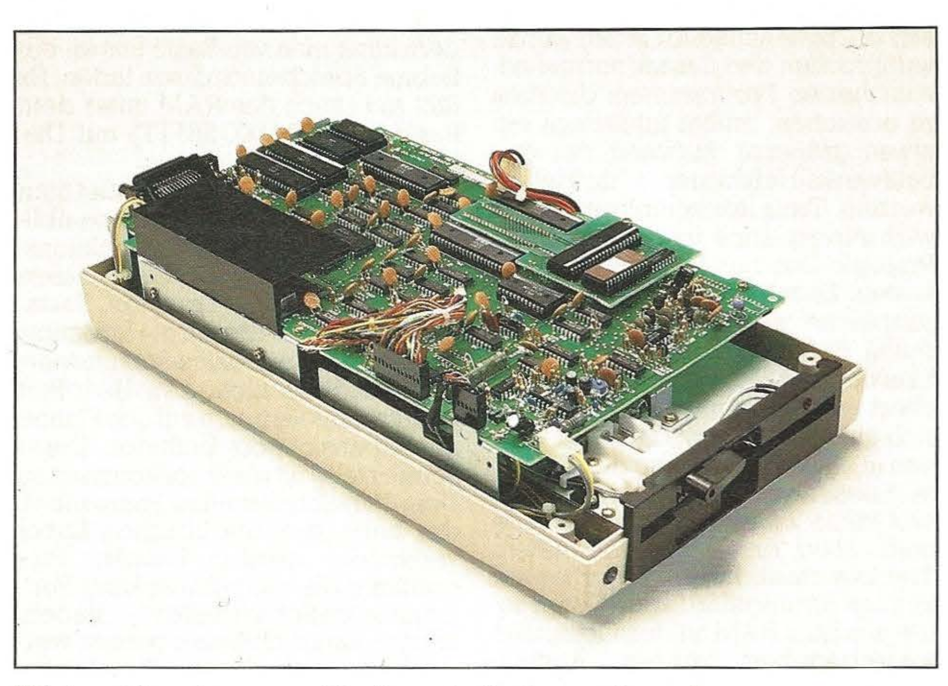


Bild. Gar nicht mehr so teuer: Die »Traum-Laufwerke« von Commodore

Der IEEE-488-Bus

Dieser Bus ist kein Commodore-typisches Konzept, er wird von verschiedenen Herstellern, vor allem für Mehrbenutzer- und Meßsysteme, verwendet. Deshalb kann man ihn durchaus als Standard-Schnittstelle für Peripheriegeräte bezeichnen. Er wurde bei nahezu allen Geräten von Commodore verwendet, die vor dem C 64 entwickelt wurden (zum Beispiel bei der 3000-, 4000-, und 8000-Serie). Sogar der legendäre Pet 2001 verfügte über diese Schnittstelle.

Darum ist es schwer verständlich, daß bei der Entwicklung des C 64 von diesem bewährten Konzept abgewichen wurde. Glücklicherweise ist der C 64 ein sehr vielseitiger Computer. So dauerte es nicht lange, bis die ersten IEEE-488-Interfaces für den C 64 auf den Markt kamen. Sehr gut eignet sich das in der 64'er, Ausgabe 7/85 vorgestellte Interface, nicht nur wegen seines Preises (Bausatz 75 Mark, Fertiggerät 98 Mark), sondern auch wegen der sehr guten Verträglichkeit mit den meisten kommerziellen Softwareprodukten. Viele Textprogramme arbeiten zum Beispiel völlig problemlos mit diesem Interface.

Es gibt eine große Palette IEEE-488-Peripheriegeräte von Commodore und anderen Herstellern. Schon die einfacheren Diskettenlaufwerke wie die 2031 oder die 4040 beeindrucken durch ihre Schnelligkeit, die fünfmal höher liegt als bei der 1541. Aber richtig interessant wird die Sache erst bei den Laufwerken SFD 1001 und 8250.

Die großen Laufwerke

Es ist schon eine beeindruckende Tatsache, wenn man nach dem Formatieren einer Diskette mit der SFD 1001 die Meldung »4133 Blocks free« bekommt. Aber dieses Laufwerk ist nicht nur fünfmal so schnell und kann mehr als das Sechsfache speichern, dieser Platz ist auch vielseitiger verwendbar, als bei der 1541 oder auch bei älteren Floppies, wie der 8050. Bei der 8050, einem Doppellaufwerk mit 500 KByte pro Diskette, dürfen relative Dateien nur zirka ein Drittel einer Diskette füllen. Diese Regelung gibt es bei der SFD 1001 nicht mehr, so daß die Größe von relativen Dateien nur durch den verfügbaren Diskettenplatz begrenzt ist. Gleichzeitig sind diese Laufwerke unerhört zuverlässig, ein kleines Beispiel aus der Praxis soll dies verdeutlichen. Beim Commodore Anwender-Club München ist eine SFD 1001 zum Betrieb der Club-Mailbox (Tel. 089/ 8120338) eingesetzt. Das Mailbox-Programm ist so konzipiert, daß es keinerlei Daten (außer dem Programm selbst) im Speicher hat. Die gesamte Datenübertragung wird direkt von und auf der SFD 1001 abgewickelt — eine enorme Belastung. Dies macht die SFD 1001 nun schon seit mehr als vier Monaten Tag für Tag, 24 Stunden durchgehend. In dieser Zeit gab es keinen einzigen Schreib- oder Lesefehler!

Alle IEEE-488-Laufwerke haben einen doppelt so großen Pufferspeicher, wie die 1541. Das bedeutet, daß drei relative oder fünf sequentielle Dateien gegenüber einer relativen oder drei sequentiellen Dateien bei der 1541 gleichzeitig geöffnet sein dürfen. Besonders interessant ist das Arbeiten mit Doppellaufwerken, wie zum Beispiel der 4040, 8050

Laufwerke für den C 64

Ein bis zwei Megabyte Speicher, fünfmal schnellere Datenübertragung und höchste Zuverlässigkeit am C 64 — warum nicht? Die »großen« Laufwerke von Commodore machen es möglich.

oder 8250. Die 4040-Floppy besitzt als einzige das gleiche Aufzeichnungsformat wie die 1541, allerdings verleiht ihr der IEEE-488-Bus fast Flügel. Trotz des gleichen Diskettenformats ist eine 4040 mehr als zwei zusammengekoppelte 1541.

Ein Doppellaufwerk ist mehr als zwei Einzellaufwerke

Durch die Befehle BACKUP und COPY dupliziert die 4040 beispielsweise eine Diskette, ohne den Computer damit zu belasten. Diese ganzen Möglichkeiten stehen natürlich auch mit der 8050- oder 8250-Floppy offen, nur mit wesentlich höheren Datenmengen. Das bedingt aber auch ein anderes Aufzeichnungsformat auf der 5¼-Zoll-Diskette. Eine 8250 kann deshalb keine auf der 1541 bespielte Disketten lesen und umgekehrt. Der einzige Weg, die Daten vom 1541 auf den SFD 1001- und 8250-Laufwerken zu verwenden, besteht darin, sie zu überspielen. Mit dem Überspielen kommerzieller Programme vom 1541- auf das 8250-Format hat es allerdings so seine Besonderheit auf sich. Obwohl die meisten Programme auch mit diesen Laufwerken (bei Verwendung des 64'er-Interfaces) problemlos funktionieren, ist man gezwungen, den, verständlicherweise angebrachten, Kopierschutz zu entfernen. Bei der Raffinesse heutiger Schutzarten wird das aber in der Regel ein erfolgloses Unterfangen sein.

DOS-Befehle

Wenn man über ein Doppellaufwerk verfügt, bekommt zum Beispiel der DOS-Befehl COPY eine ganz neue Bedeutung. Mit ihm kann man jetzt einzelne Files von einer Diskette auf eine andere kopieren. Man kann aber auch alle Möglichkeiten der Namensauswahl treffen, die schon von der 1541 her bekannt sind. Damit sind die Zeichen »?« und »*«, die die Programmauswahl aus dem Disketteninhaltsverzeichnis wesentlich erleichtern, gemeint. So kann man jetzt zum Beispiel alle Programme, die mit »C« anfangen, mit dem Befehl: OPEN1,8,15,»C0:* = 1:C*« leicht vom Laufwerk 1 zum Laufwerk 0 kopieren. Übrigens kann man mit dem COPY-Befehl auch Dateien beim Kopieren zusammenfügen, wie es ja schon von der 1541 her bekannt ist. Das geht aber eigentlich nur bei sequentiellen Dateien gut, da Programme nur physisch zusammengehängt werden. Was weniger bekannt ist, und auch bei der 1541 geht, ist das Anhängen einer Datei an eine andere, ohne die erste Datei zu kopieren. Der Befehl lautet hier: »C0:FILE1 = 0:FILE1,0:FILE2«.

Das bedeutet, daß die Datei mit dem Namen »FILE2« ein zweites Mal auf der Diskette angelegt und an die Datei »FILE1« angehängt würde. Die Laufwerksbezeichnungen »0:« sind mit anzugeben, obwohl sie eigentlich überflüssig wären. Werden sie weggelassen, so meldet die Floppy den Fehler »FILE EXISTS« und deutet damit an, daß sie die Datei mit Namen »FILE1« auch kopieren will, aber der Zielname identisch mit dem Urnamen ist. Für Doppellaufwerke interessant ist der Befehl »DUPLICATE«, mit dem die Floppy eine komplette Kopie aller Informationen durchführt. Dieser Befehl wird in den Handbüchern von Commodore nicht erläutert. Deshalb sei hier näher darauf eingegangen. Die Syntax des Befehls lautet »D:1 = 0«.

Hierbei findet eine Datenübertragung einer gesamten Diskette vom Laufwerk 0 auf das Laufwerk 1 statt. Es geht natürlich auch umgekehrt, wenn man die beiden Werte vertauscht. Die Floppy formatiert die angegebene Zieldiskette erst und kopiert dann Spur für Spur von der Original- auf die Diskette. Dieser Vorgang läuft intern in der Floppy ab, belastet also weder den Bus noch den daran angeschlossenen Computer.

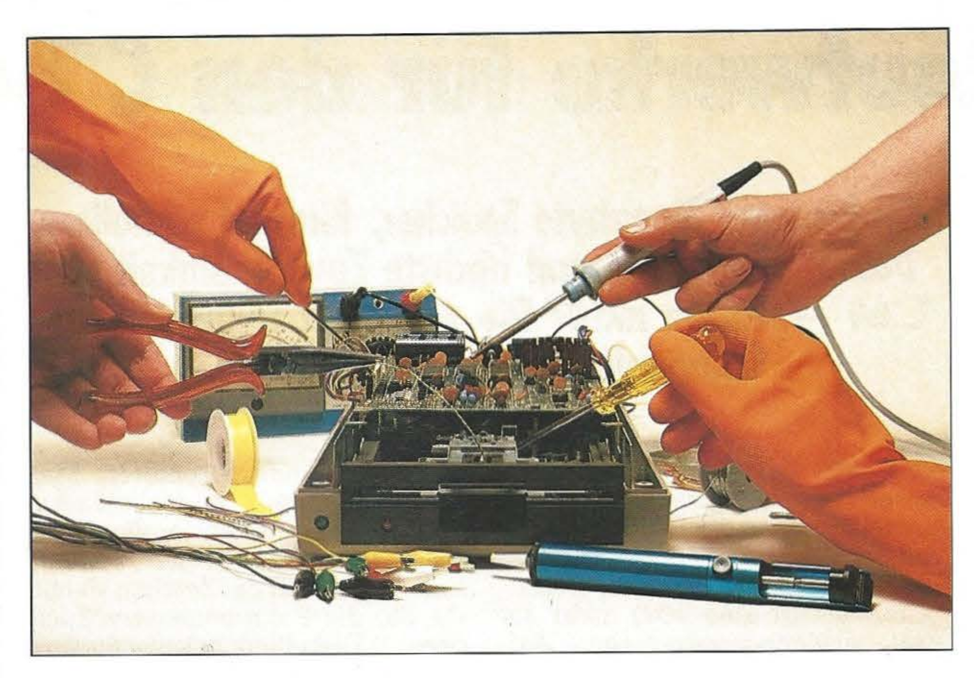
Endlich ein angemessener Massenspeicher

Insgesamt betrachtet, kann man sagen, daß das Arbeiten mit den »großen« Floppies von Commodore einfach fantastisch ist. Man hat endlich einen Massenspeicher, der dem C 64 gewachsen ist, sowohl in bezug auf die Geschwindigkeit als auch den verfügbaren Speicherplatz. Der Star dieser Laufwerke (abgesehen vom Festplattenlaufwerk mit 9 MegaByte) ist die 8250, denn sie besteht aus zwei Laufwerken mit insgesamt vier Schreib-/Leseköpfen. Mit ihr kann man 2 MegaByte direkt ansprechen. Das ist wesentlich mehr als der Marktführer der Personal Computer, der IBM-PC, bieten kann. Dessen Laufwerke haben in der Grundausstattung nur zweimal 360 KByte und damit kaum mehr als ein Drittel der Kapazität einer 8250 anzubieten.

Warum eigentlich nicht?

Die 1541/1571 wird sicherlich das Standard-Laufwerk für den C 64 bleiben. In ihrem Disketten-Format gibt es die meisten Programme und ihre Leistungen reichen in der Mehrzahl der Anwendungen auch aus. Da sich aber auch im Heimbereich Anwendungen wie Datenverwaltung und Textverarbeitung immer mehr durchsetzen, ist die Anschaffung eines »großen« Laufwerks sicherlich eine Überlegung wert. Eine der interessantesten Alternativen ist die SFD 1001, die, zusammen mit dem von uns veröffentlichten Interface, weniger als 800 Mark kostet. Nicht zu unterschätzen ist auch der Preisvorteil bei den Speichermedien, denn immerhin paßt auf eine mit der SFD bespielten Diskette das Sechsfache einer 1541-Diskette. In Kombination mit diesen Speicherriesen macht der C 64 fast seinem Nachfolger Konkurrenz, dem C 128.

(K. Hinsch/aw)



Diskettenlaufwerk 1541 selbst justiert

Es ist bekannt, daß das mechanisch anfällige 1541-Diskettenlaufwerk häufig Probleme beim Lesen eigener und im besonderen fremder Disketten hat. Hier werden zwei Verfahren vorgestellt, mit denen Sie diesen Fehler beseitigen können.

In den meisten Fällen ist die Ursache ein verstellter Schreib-Lesekopf. Wird das Laufwerk zu einem Reparaturservice eingeschickt, sind Wartezeiten bis zu drei Monaten leider üblich. Außerdem ist die Reparatur nicht ganz billig, und oft tritt der gleiche Fehler innerhalb kürzester Zeit wieder auf.

Wir wollen Ihnen deshalb zwei Verfahren vorstellen, mit denen Sie das 1541-Diskettenlaufwerk selbst einstellen können, beziehungsweise von jedem Radio und Fernsehfachhandel einstellen lassen können. Bei dem ersten Verfahren handelt es sich um eine reine Softwarelösung, mit der das Laufwerk zwar nicht exakt eingestellt werden kann, in den meisten Fällen ist es aber vollkommen ausreichend. Mit dem zweiten Verfahren kann das Laufwerk dagegen absolut genau eingestellt werden. Benötigt wird allerdings ein Oszilloskop.

Fixierung der Mechanik

Unabhängig vom Verfahren muß zuerst der Schreib-Lesekopf neu fixiert werden. Schalten Sie dazu bitte alle Geräte aus und ziehen das Netz- und IEC-Kabel aus dem Laufwerk.

Nach dem Lösen der sechs Kreuzschlitzschrauben auf der Unterseite, läßt sich der Gehäusedeckel des Laufwerks entfernen. Da sich der Schrittmotor, an dem später die Einstellung vorgenommen wird auf der Chassisunterseite befindet, muß auch der Gehäuseboden entfernt werden, der mit sechs Kreuzschlitzschrauben befestigt ist. Lösen Sie bitte diese sechs Schrauben und ziehen vorsichtig den Stecker ab, der die grüne Leuchtdiode mit der Platine verbindet. Um später beim Zusammenbau Fehler zu vermeiden, markieren Sie alle Steckverbindungen mit einem Filzstift. Als nächstes lösen Sie alle Schrauben, die die Platine mit dem Chassis verbinden und ziehen vorsichtig die restlichen Steckverbindungen ab.

Die Platine kann nun abgehoben und zur Seite gelegt werden. Liegt das »nackte« Chassis vor Ihnen, sehen Sie auf der rechten Seite ein Stahlband, das den Schreib-Lesekopf mit dem Schrittmotor über einen Zylinder verbindet (Bild 1). Da der Zylinder sich bei Erwärmung mehr ausdehnt als die Achse des Schrittmotors und nur über eine Preßpassung mit dieser verbunden ist, tritt zwischen Schrittmotor-Achse und Zylinder ein Schlupf auf. Das heißt, daß sich der Zylinder auf der Achse beim Formatieren einer neuen Diskette verdreht und dadurch den Schreib-Lesekopf verstellt. Um das zu verhindern, muß die Achse und der Zylinder im betriebswarmen Zustand verklebt werden. Dazu eignen sich Superkleber, wie sie von vielen Firmen angeboten werden oder Epoxydharz, das mit einem Härter versehen ausreichend Festigkeit gewährleistet. Lassen Sie diesen Kleber vorsichtig zwischen Achse und Zylinder laufen. Passen Sie aber auf, daß der Kleber nicht versehentlich mit dem Stahlband in Berührung kommt. Nach etwa drei Stunden Trockenzeit können Platine und Chassis zusammengeschraubt werden. Vergessen Sie nicht die Steckverbindungen wiederherzustellen. Der Schreib-Lesekopf ist jetzt fixiert, und der eigentliche Einstellvorgang kann beginnen. Stellen Sie dazu das Laufwerk so vor sich hin, daß der Diskettenschacht senkrecht steht. Stellen Sie die Verbindung mit dem Computer und dem Netz wieder her und schalten alle Geräte ein.

Verfahren 1 (Softwarelösung)

Laden Sie das vorher abgetippte Programm (Listing 1) und legen eine »sauber« bespielte Diskette in das Laufwerk (am besten die Test Demo Diskette). Nach dem Lösen der zwei Schrittmotor-Befestigungsschrauben (Einstellschrauben) an der Unterseite des Laufwerks, kann das Programm mit RUN gestartet werden. Der Menüunterpunkt »Feineinstellung« eignet sich nicht zur Justierung des Diskettenlaufwerkes; denn hier werden alle Spuren getestet, und die Lesezeiten der einzelnen

```
100 REM ******  FLOPPY - ADJUST ******  :  <233>
120 PRINT"{CLR,CTRL-N,WHITE}":POKE 53280,1
    2:POKE 53281,0:POKE 198,0              <166>
140 G=1:MG=3:R1=10:R0=3:RV=0:A$="{10SPACE}
    "                                      <142>
150 CU=214:REM CURSORZEILE                 <126>
160 PRINT"{HOME,2DOWN}"A$"FLOPPY-EINSTELLH
    ILFE"                                  <120>
170 PRINT"{3DOWN}"                         <169>
180 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    @@@@@@@@";                             <165>
190 IF C/R1=INT(C/R1)THEN RV=NOT RV        <178>
192 IF RV THEN PRINT"{RVSON}";             <146>
195 PRINT"{LIG.BLUE,8SPACE}BITTE MIT 'CRSR
    ' WAEHLEN{8SPACE}";                    <154>
196 PRINT"{8SPACE}DANN{2SPACE}'RETURN'{2SP
    ACE}DRUECKEN{8SPACE,WHITE}";:POKE 199,
    0                                      <126>
197 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
    TTTTTTTT"                              <074>
200 X=1:W$="GROBEINSTELLUNG":GOSUB 400     <067>
220 W$="FEINEINSTELLUNG":GOSUB 400         <112>
240 W$="{4SPACE}ENDE{7SPACE}":GOSUB 400    <223>
280 GET G$:IF G$=CHR$(13)THEN 360          <198>
300 IF G$="{DOWN}"THEN G=G+1:IF G>MG THEN
    G=1                                    <159>
320 IF G$="{UP}"THEN G=G-1:IF G<1 THEN G=M
    G                                      <114>
330 POKE 198,PEEK(198)AND 1                <059>
340 GOTO 160                               <110>
360 ON G GOSUB 445,445:ON G GOTO 460,500,4
    50                                     <000>
380 GOTO 280                               <190>
400 PRINT:PRINT A$;:IF X=G THEN PRINT"{RVS
    ON}";                                  <152>
420 PRINT W$"{RVOFF,2SPACE}";              <060>
430 IF X=G THEN C=C+1 AND 15:IF C/R0=INT(C
    /R0)THEN PRINT"{LEFT}←";               <134>
440 PRINT:X=X+1:RETURN                     <022>
445 POKE CU,24:PRINT:FOR F=1 TO 11:PRINT:N
    EXT:POKE CU,10:PRINT                   <217>
446 POKE 768,174:POKE 769,167:POKE 144,0:O
    PEN 1,8,15:CLOSE 1:REM TEST DEVICE PRE
    SENT                                   <080>
447 POKE 768,139:POKE 769,227              <061>
448 IF ST=0 THEN RETURN                    <146>
449 E$="FLOPPY NICHT EINGESCHALTET":GOTO 1
    340                                    <187>
450 CLOSE 2:CLOSE 1:PRINT"{CLR,SPACE}BYE."
    :END                                   <076>
460 REM GROB                               <228>
480 T1=18:T2=18:GOTO 560                   <159>
500 REM FEIN                               <118>
520 PRINT"LESEN VON SPUR ";:INPUT" 1{3LEFT
    }";T1                                  <134>
540 PRINT"LESEN BIS SPUR ";:INPUT" 35{4LEF
    T}";T2                                 <078>
560 INPUT "MAX. ZEIT FUER 1 BLOCK{3SPACE}0
    .5 SEC{9LEFT}";G$:MT=VAL(G$)           <104>
570 FB=0:INPUT"RANDFARBWECHSEL GEWUENSCHT<
    J/N>{2SPACE}N{3LEFT}";G$:IF G$="J"THEN
     FB=1                                  <127>
580 PRINT"{CLR}BITTE KORREKT FORMATIERTE D
    ISKETTE"                               <246>
600 PRINT"EINLEGEN.(DATEN WERDEN{SPACE,RVS
    ON}NICHT{RVOFF,SPACE}ZERSTOERT)"       <106>
610 PRINT:PRINT"NACH DIESER DISKETTE KANN
    DIE FLOPPY":PRINT"EINGESTELLT WERDEN"  <135>
620 PRINT:PRINT"FERTIG ? DANN TASTE DRUECK
    EN"                                    <218>
640 POKE 198,0:WAIT 198,1:GET G$           <068>
680 ZL=10:SP=12:REM POS. F.BALKEN          <057>
700 DL=59903:REM ADR.DELETE LINE           <212>
720 XR=781:REM X-REG.SYS                   <168>
740 LB=20:B$="{RVSON}":FOR F=1 TO LB:B$=B$
    +" ":NEXT                              <244>
780 DEF FN T(X)=INT(PEEK(X)/16)*10+PEEK(X)
    -INT(PEEK(X)/16)*16                    <062>
800 Z=56328:S=Z+1:POKE S,0:POKE Z,0        <099>
820 PRINT"{CLR,DOWN,2SPACE}****{2SPACE}FLO
    PPY-ADJUST{2SPACE}****{2SPACE}V25.2.85
    "                                      <044>
840 :                                      <054>
860 POKE CU,5:PRINT:PRINT" DISK WIRD INITI
    ALISIERT"                              <115>
880 OPEN 1,8,15,"I":OPEN 2,8,2,"#"         <109>
900 POKE CU,ZL:PRINT                       <005>
920 PRINT TAB(SP)" ↑ MAX."SPC(LB-13)"MIN.
    ↑"                                     <102>
940 T=T1:REM TRACK                         <035>
960 POKE S,0:POKE Z,0                      <081>
980 PRINT#1,"U1:"2;0;T;1                   <036>
1000 INPUT#1,E,E$,ET,ES:IF E THEN 1340     <047>
1040 TM=FN T(S)+FN T(Z)/10-D               <059>
1060 FL=NOT FL:IF FL AND FB THEN POKE 5328
     0,(PEEK(53280)+1)AND 1:REM ALLE FARBE
     N-AND15                               <121>
1080 TB=LB-LB*(TM/MT):IF TB<0 THEN TB=0    <233>
1100 POKE XR,ZL:SYS DL                     <196>
1120 PRINT"ZEIT :"TM"S"TAB(SP)LEFT$(B$,TB+
     1)                                    <049>
1140 POKE CU,5:PRINT:PRINT"{RVSON,LIG.BLUE
     }EINSTELLUNG{2SPACE}O.K{5SPACE,RVOFF,
     WHITE}";                              <098>
1160 PRINT"SPUR : "T"{LEFT,2SPACE}"        <201>
1180 IF TM>MT THEN PRINT"{RVSON,UP,RED}FAL
     SCHE EINSTELLUNG !{RVOFF,WHITE}"      <001>
1200 TX=T:T=T+1:IF T>T2 THEN T=T1          <191>
1220 D=ABS(T-TX)/100:REM FUER TRACKTRANSPO
     RT ZEIT ZUGEBEN                       <049>
1240 GET G$:IF G$<>""THEN 1380             <163>
1260 GOTO 960                              <076>
1340 PRINT"{CLR,RVSON}FLOPPY - FEHLER{RVOF
     F,SPACE}!{5DOWN}":PRINT E;"{RVSON}"E$
     "{RVOFF}";ET;ES                       <228>
1360 PRINT:PRINT"BITTE TASTE DRUECKEN.":PO
     KE 198,0:WAIT 198,1                   <082>
1380 CLOSE 2:CLOSE 1:RUN                   <043>
```


Listing 1. Mit diesem Programm können Sie das 1541-Diskettenlaufwerk justieren. Bei der Eingabe bitte Seite 54 beachten

Spuren weichen voneinander minimal ab. Es läßt sich also keine minimale Lesezeit einstellen. Wählen Sie deshalb den Menüunterpunkt Grobeinstellung. Sie werden nach der maximal zulässigen Lesezeit gefragt. Die Voreinstellung beträgt 0,2 Sekunden. Das ist die Zeit, die dem Laufwerk zur Verfügung steht, um einen Sektor zu lesen. Wird diese Zeit überschritten, erscheint auf dem Bildschirm in roter Farbe eine Fehlermeldung. Ist die Lesezeit kürzer als 0,2 Sekunden wird einmal die tatsächliche Zeit, die das Laufwerk braucht um einen Sektor zu lesen angezeigt. Zum anderen wird diese Lesezeit optisch durch einen querliegenden Balken verdeutlicht.

Versuchen Sie durch Verstellung des Schrittmotors diese Lesezeit auf ein Minimum zu justieren. Ist das geschehen, sind die Einstellschrauben leicht anzuziehen, so daß sich der Motor nicht mehr von selbst verstellen kann. Sie werden sehen, daß sich die Lesezeit alleine durch das Festschrauben verändert hat. Versuchen Sie jetzt, indem Sie den Vorgang wiederholen, die Lesezeit nochmals zu optimieren. Anschließend sind die Einstellschrauben fest anzuziehen und das Laufwerk wie oben beschrieben, nur in umgekehrter Reihenfolge, zusammenzubauen.

Verfahren 2 (Hardwarelösung)

Da der C 64 wie jeder andere Computer auch, ein Gerät ist, das Daten digital verarbeitet, lassen sich analoge Einstellvorgänge zwar hinreichend genau, aber nicht exakt durchführen. Aus diesem Grund haben wir eine »Hardwarelösung« entwickelt, mit der jeder, der den Umgang mit einem Oszilloskop be-

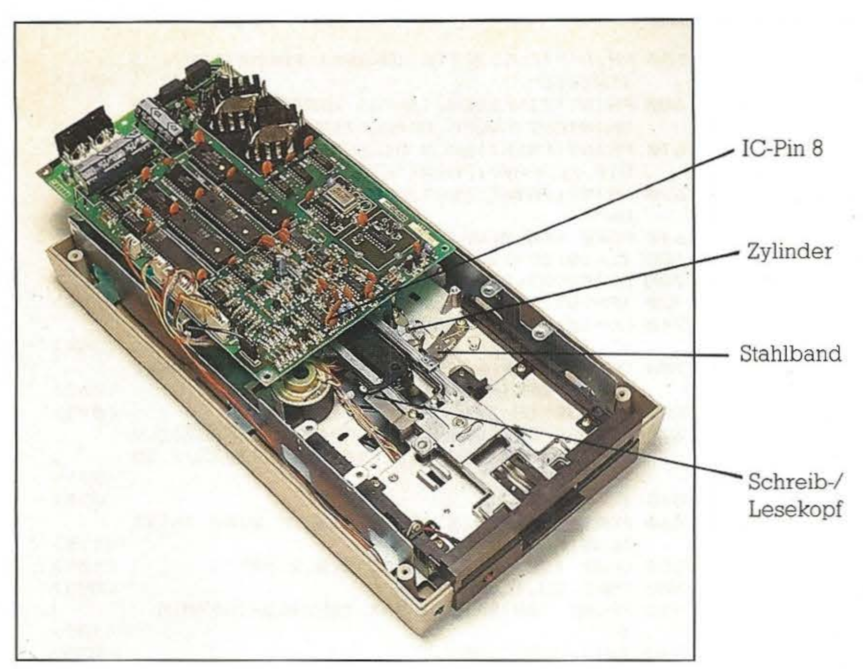


Bild 1. Hier finden Sie die wichtigsten Punkte zur Einstellung des Diskettenlaufwerks mit einem Oszilloskop

herrscht, das 1541-Diskettenlaufwerk absolut exakt einstellen kann. Voraussetzung ist natürlich auch wieder eine Diskette, die mit einem Disketten-Laufwerk beschrieben wurde, das exakt eingestellt war. Die Vorgehensweise ist die gleiche wie beim Verfahren 1, mit dem Unterschied, daß nach dem Einschalten aller Geräte und nach dem Lösen der Einstellschrauben folgende Zeilen anstelle des Programms eingegeben werden müssen:

```
10 OPEN 2,8,2,"#": OPEN 15,8,15
20 PRINT# 15,"U1 2 0 18 0":GOTO 20
```


Die Aufgabe dieses kleinen Programms, das mit RUN gestartet wird, ist das permanente Lesen des Sektors 0 auf Spur 18. Diese Spur und dieser Sektor sind mit Absicht so gewählt worden, da der Abstand nach Spur 1 und Spur 35 der gleiche ist. Stellen Sie nun die Y-Ablenkung des Oszilloskops auf 10 mV/cm und die X-Ablenkung auf 1 mS. Verbinden Sie den Tastkopf, der bei den angegebenen Einstellwerten ein Tastverhältnis von 1/10 haben muß, mit dem in Bild 1 markierten IC-Pin 8. Das Oszillogramm zeigt ein »hochfrequentes« Signal, das bei exakter Justierung einen Spannungspegel von etwa 350 mV_{ss} haben sollte. Dieser Spannungspegel ist aber nur ein Anhaltspunkt und hängt in erster Linie vom Alter der Diskettenaufnahme ab. Verdrehen Sie den Schrittmotor so lange, bis der Spannungspegel ein Maximum erreicht. Auch ist bei einem dejustiertem Schreib-Lesekopf dem hochfrequenten Signal ein niederfrequentes Signal überlagert, das an der zyklischen Veränderung der Amplitude zu erkennen ist. Dieses überlagerte Signal verschwindet bei exakt eingestelltem Schreib-Lesekopf.

(M. Kastermeier/ah)

Die Datasette streikt nie wieder

Einer der häufigsten Fehler, der bei der Datasette auftritt, ist ein verstellter Tonkopf. Dieser Fehler macht sich besonders dann bemerkbar, wenn mit Turbo Tape oder ähnlichen Programmen gearbeitet wird. Mit der hier beschriebenen Schaltung läßt sich extrem einfach, ohne jegliches Programm, der Tonkopf an jede Datenkassette anpassen.

Um die Datasette oder einen anderen Datenrecorder zu justieren, gibt es grundsätzlich zwei Möglichkeiten. Eine kleine elektronische Schaltung, mit der sich unabhängig vom Computer die Tonkopfstellung an jede Datenkassette anpassen läßt und ein Programm, das in irgendeiner Form die Tonkopfstellung grafisch auf dem Monitor des Computers darstellt. Ein solches Programm ist aber unbrauchbar, egal wie gut oder schlecht es ist. Der Grund dafür ist ganz einfach der, daß sich nach erfolgter Justage Programme, die zuvor auf anderen Kassetten gespeichert wurden, nicht mehr laden lassen; unter anderem auch das Justageprogramm selbst. Sollen solche Programme geladen werden, müßte das Justageprogramm noch einmal abgetippt werden.

Um das zu vermeiden, stellen wir Ihnen eine Schaltung vor, mit der das Einstellen extrem einfach wird.

Damit die Schaltung verständlich wird, zuerst ein paar Worte zur Datasetten-Elektronik.

Sie besteht aus zwei Hauptgruppen, einem zweistufigen Verstärker, der die Aufgabe hat, das analoge Signal, das vom Tonkopf kommt, zu verstärken. Analog deshalb, weil sich digitale Signale nicht auf Band speichern lassen. Selbst wenn ein solches Signal am Tonkopf anliegt, wird es nicht als solches auf das Band geschrieben, sondern in Form einer Sinusschwingung. Beim Laden muß diese Sinusschwingung wieder in eine Form gebracht werden, die der Computer versteht. Also Rechtecksignal. Dies geschieht in der zweiten Hauptstufe, mit Hilfe eines Schmitt-Triggers. Am Ausgang des Schmitt-Triggers liegt das Signal in Form einer Rechteckschwingung vor, die entweder einen Spannungspegel von 0 oder 5 Volt hat. Dieses Signal eignet sich nicht zur Einstellung des Tonkopfes, weil die Ampli-

R = 270 Ω
D = LED
IC = LF 356
Tr. = 10 kΩ

Bild 1. Schaltung des Komperators

5 V Betriebsspannung
Komparator-ausgang
Schwellwert in die Nähe des Scheitelwerts
Schwellwert
Signal
D = dunkel
H = hell

Bild 2. Die Helligkeit der Leuchtdiode ist abhängig von der Zeitspanne, die der Komperatorausgang auf +5V liegt.

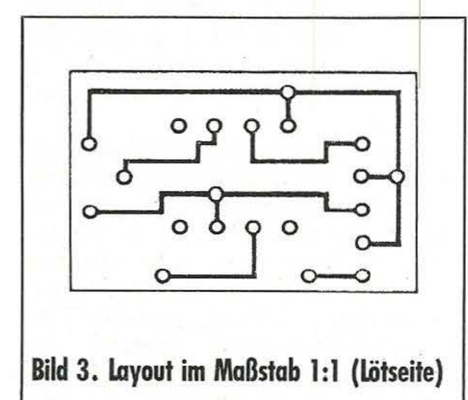


Bild 3. Layout im Maßstab 1:1 (Lötseite)

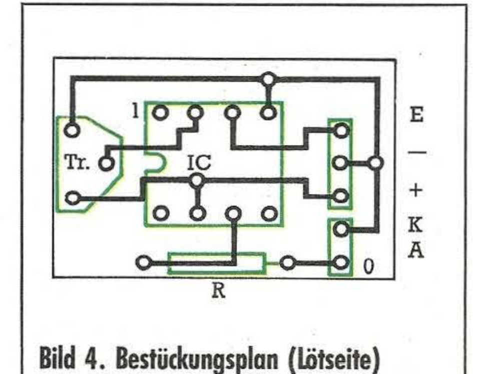


Bild 4. Bestückungsplan (Lötseite)

tude des Signals, unabhängig von der Tonkopfstellung, immer konstant zwischen 0 und 5 Volt hin- und herspringt.

Die Messung mit einem Oszilloskop ergab aber, daß, abhängig von der Tonkopfstellung, die Amplitude der analogen Spannung schwanke. Ist der Tonkopf optimal eingestellt, geht die Amplitude der Spannung gegen ein Maximum. Ist der Tonkopf dejustiert, weicht die Amplitude, abhängig von der Tonkopfstellung, vom Maximum ab. Man kann es jedoch keinem Datasetten-Besitzer zumuten, sich ein Oszilloskop anzuschaffen, nur um die Datasette zu justieren.

Bastelanleitung

Die vorliegende Bastelanleitung, deren Bauteile zu einem Preis von unter fünf Mark zu haben sind, ersetzt in diesem Fall ein Oszilloskop. Mit der Schaltung (Bild 1) läßt sich eine Spannung, natürlich in gewissen Grenzen, auf Maximum abgleichen. Das Herz ist ein Operationsverstärker vom Typ LF 356, der als Komparator (Schwellwertschalter) betrieben wird. Außerdem hat dieser Operationsverstärker gegenüber anderen den Vorteil, daß seine Eingangsstufe aus einem Feldeffekttransistor besteht. Der Eingangswiderstand geht dadurch gegen unendlich und belastet das zu messende Signal in keinster Weise. Mit dem Trimmpotentiometer läßt sich eine Schwellspannung (Bild 2) einstellen, die laufend mit der analogen Sinusschwingung verglichen wird.

Ist der Momentanwert der Sinusschwingung kleiner als die vorgegebene Schwellspannung, führt der Ausgang des LF 356 0 Volt. Wird der Momentanwert größer, springt der Ausgang des LF 356 auf +5 Volt und regt dadurch eine Leuchtdiode an. Wird die Schwellspannung in den Scheitelpunkt der Sinusschwingung gelegt (gestrichelte Linie in Bild 2), geht die Zeitspanne, in der der Ausgang des Komperators auf 5 Volt liegt, gegen ein Minimum. Daraus folgt, daß die Helligkeit der Leuchtdiode abnimmt, je näher die Schwellspannung an den Scheitelwert der Sinusschwingung rückt. Wird dagegen die Amplitude des Signals, also der Sinusschwingung, vergrößert, wird die Helligkeit der Leuchtdiode wieder größer. Denn die Zeitspanne, in der der Ausgang des Komparators auf 5 Volt liegt, wird größer. Dieses ist vom Prinzip

her der ganze Abgleichvorgang. Mit dem Trimmpotentiometer wird auf minimale Helligkeit und mit der Tonkopfeinstellschraube auf maximale Helligkeit abgeglichen.

Aufgebaut wird die Schaltung auf einer kleinen Lochrasterplatine. Diejenigen, die sich eine Platine ätzen wollen, finden das Layout im Verhältnis 1:1 in Bild 3. Wie die einzelnen Pins der Bauelemente miteinander verbunden werden, zeigt Bild 4. Achten Sie beim Zusammenbau auf die richtige Polarität der Leuchtdiode (Bild 5).

Ist die Schaltung zusammengelötet, muß sie noch im Datasettengehäuse untergebracht werden. Öffnen Sie dazu die Datasette und bohren an einer geeigneten Stelle ein Loch in das Gehäuseoberteil, so daß die Leuchtdiode gerade in dieses Loch paßt.

Verbinden Sie die Anschlüsse »+« und »—« (Bild 4) mit den Motoranschlußklemmen. Dabei ist ebenfalls auf die Polarität zu achten. Im allgemeinen ist sie auf dem Motor gekennzeichnet. Der in Bild 4 gekennzeichnete Punkt »E« (für Eingang) muß über ein abgeschirmtes Kabel mit einem der beiden Lötpunkte auf der Datasettenplatine (Bild 6) verbunden werden. Die Abschirmung ist an den mit »—« gekennzeichneten Punkt (Bild 4) zu löten.

Bei den beiden Lötpunkten handelt es sich um den Ausgang des ersten beziehungsweise zweiten Analogverstärkers einer Commodore-Datasette.

Geräte anderer Hersteller sind zum Teil anders aufgebaut. Es kann vorkommen, daß die beiden in Bild 6 gekennzeichneten Analogverstärker in einem Gehäuse untergebracht sind. In diesem Fall ist der Punkt »E« mit dem Pin 8 dieses ICs zu verbinden.

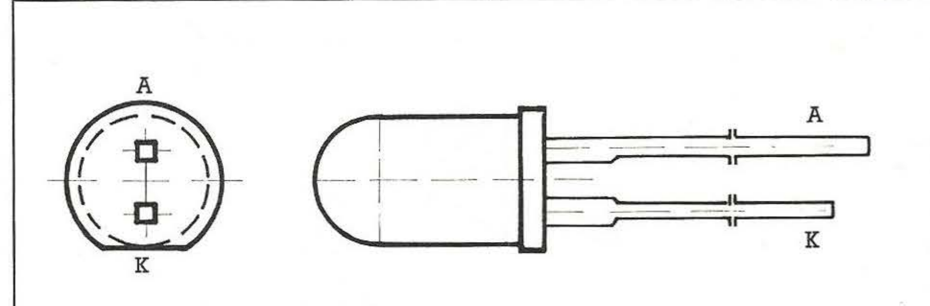


Bild 5. Beim Einlöten der Leuchtdiode unbedingt auf die Polarität achten. Anode=A=längeres Beinchen oder runde Seite.

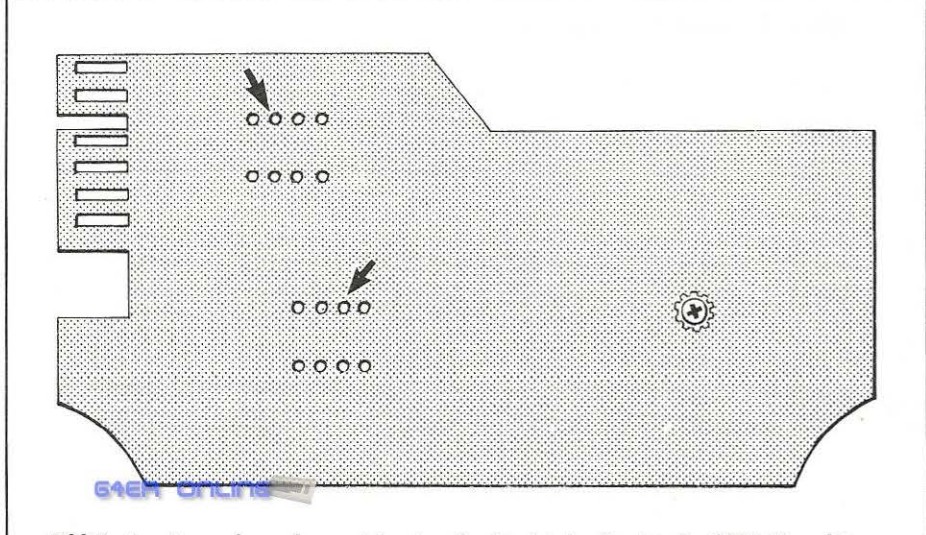


Bild 6. An einem der gekennzeichneten Punkte ist der Punkt »E« (Bild 4) zu löten.

Einfaches Einstellen

Bevor die Datasette zusammengebaut wird, ist die Schaltung an die Datasetten-Elektronik anzupassen. Schalten Sie dazu den C 64 ein, legen eine Programm-Kassette in die Datasette und drücken die PLAY-Taste.

Nach der direkten Eingabe POKE 54272+24,15 drehen Sie den Lautstärkeregler Ihres Monitors auf Maximum und warten bis die Übertragung des Programm- oder Datencodes im Lautsprecher zu hören ist. Nun muß in einem wechselseitigen Einstellvorgang die Helligkeit der Leuchtdiode am Trimmpotentiometer auf Minimum und an der Tonkopfeinstellschraube auf Maximum abgeglichen werden. Bei Commodore-Datasetten befindet sich die Tonkopfeinstellschraube (Kreuzschlitz) bei gedrückter PLAY-Taste unter einem etwa 5 mm großen Loch auf dem Gehäuseoberteil. Soll eine Datasette justiert werden, muß der Kassettendeckel abgebaut werden. Die Tonkopfschraube ist nun eine der beiden Tonkopfbefestigungsschrauben und zwar die, an der sich eine Spiralfeder befindet. Doch nun zum Abgleichvorgang. Dazu gehen Sie bitte folgendermaßen vor:

1. Am Trimmpotentiometer drehen, bis die Leuchtdiode schwach flackert. Dadurch wird die Schwell- oder Schaltspannung in den Scheitelpunkt der Sinusschwingung gelegt.
2. An der Tonkopfeinstellschraube drehen, bis die Helligkeit der Leuchtdiode ein Maximum erreicht hat. Dadurch wie die Amplitude des Signals, das vom Tonkopf kommt, auf Maximum abgeglichen.

Der letzte Punkt ist nur dann erforderlich, wenn die Datasette nicht optimal eingestellt war, beziehungsweise eine Kassette benutzt wird, die mit einer anderen Datasette beschrieben wurde. In diesem Fall muß der Einstellvorgang solange wiederholt werden, bis eine Einstellung erreicht ist, bei der die Leuchtdiode erlischt sobald der Tonkopf minimal verstellt wird. Bauen Sie nun die Datasette wieder zusammen. Schalten Sie voher aber den C 64 aus.

Wollen Sie jetzt ein Programm laden, das mit einem dejustiertem Tonkopf aufgenommen wurde, brauchen Sie nur noch, nachdem der C 64 eingeschaltet wurde, die Kassette einzulegen, die PLAY-Taste zu drücken und so lange an der Tonkopfeinstellschraube drehen, bis die Helligkeit der Leuchtdiode ein Maximum erreicht hat. (ah)

Das große Rennen

Neben der Floppy und der Datasette gibt es noch die sogenannten Bandlaufwerke. Das bekannteste aus dieser Gruppe schickten wir zu einem Geschwindigkeitsvergleich auf die Rennbahn.

Daß Geschwindigkeit keine Hexerei ist, beweisen die inzwischen recht zahlreichen Hard- und Software-Erweiterungen, die die Floppy schneller machen sollen. Kein Zweifel, Schnelligkeit ist gefragt. Lange Wartezeiten beim Laden sind out.

Bei diesen Worten denkt man natürlich sofort an die »Speicherschnecke« Datasette. Aber nicht jeder hat das Geld, ein Floppy-Laufwerk zu kaufen; es muß also ein Kompromiß geschlossen werden. Die eine Möglichkeit ist, das »Turbo-Tape de Luxe« aus dem 64'er-Sonderheft Nummer 5 abzutippen, die andere, sich eine Floppy-Alternative anzuschaffen. Damit meinen wir Bandlaufwerke mit besonderen Kassetten. Den bekanntesten Vertreter dieser Klasse, nämlich das »Quick-Data-Drive« von NCS, haben wir einmal der Datasette, dem Turbo-Tape und der Floppy in einem Geschwindigkeitsvergleich gegenübergestellt.

Testbedingungen:

Es galt erstens, ein genau 8 KByte langes Basic-Programm zu speichern und wieder zu laden, und zweitens, eine sequentielle Datei mit 2 KByte Länge ebenfalls zu speichern und wieder zu laden.

Beim Arbeiten mit dem Quick-Data-Drive (QDD) zeigte sich schon bald eine deutliche Schwäche: Um das Gerät steuern zu können, muß ein spezielles »Quick-Operating-System« in den Computer geladen werden. Dieses Maschinenprogramm steht ab $C000 (49152) im Speicher. Da dieser Speicherbereich aber von vielen Basic-Programmen als Raum für Maschinenunterroutinen benutzt wird, kann es zu umfangreichen Problemen kommen.

Ein weiteres Problem: Da das QDD spezielle Endloskassetten verwendet, muß es bei jedem Laden das gesamte Band durchspulen, bis es die richtige Stelle gefunden hat. Bei den längsten Bändern (128 KByte) kann dies durchaus zu deutlichen Wartezeiten führen.

In der ersten Disziplin (Basic-Programme) zeigte sich schon, daß die Datasette, mit Turbo-Tape beschleunigt, durchaus der Floppy die Stirn bieten kann (siehe Tabelle).

Die zweite Überraschung bot das QDD: Es war fast doppelt so schnell wie das Diskettenlaufwerk! Beim Programme-Laden benötigte die Floppy 1541 sogar mehr als dreimal soviel Zeit.

Als zweite Disziplin waren sequentielle Dateien angesagt.



Quick-Data-Drive. Eine sinnvolle Anschaffung?

Auch dieser Durchgang verlief nicht ohne Schwierigkeiten: Turbo-Tape unterstützt nämlich die sequentielle Dateiverwaltung nicht.

Auch das QDD zeigt sich hier von seiner schwachen Seite. Die Floppy ist bei der Arbeit mit sequentiellen Dateien fast doppelt so schnell wie das Bandlaufwerk.

Als Fazit unseres kleinen Vergleichstests kann man sagen, daß das Quick-Data-Drive nur für den reinen Basic-Programmierer geeignet ist. Die Vorteile beschränken sich auf hohe Übertragungsgeschwindigkeit und einen relativ niedrigen Anschaffungspreis von 198 Mark. Die gravierendsten Nachteile sind, daß es erstens kaum fertige Software für dieses Laufwerk zu kaufen gibt und zweitens wegen der sehr ungünstig gewählten Speicherbelegung praktisch keine kommerziellen Programme mit dem QDD lauffähig sind. Auch sind die Kassetten mit Preisen zwischen 7,80 Mark für das 16-KByte-Band und 10,80 Mark für das 128-KByte-Band nicht gerade billig. (tr)

Info: Quick-Data-Drive, Nettetaler Computer Shop (NCS), Steylerstr. 22, 4054 Nettetal 2, Tel. 02157/1616, Preis: 198 Mark

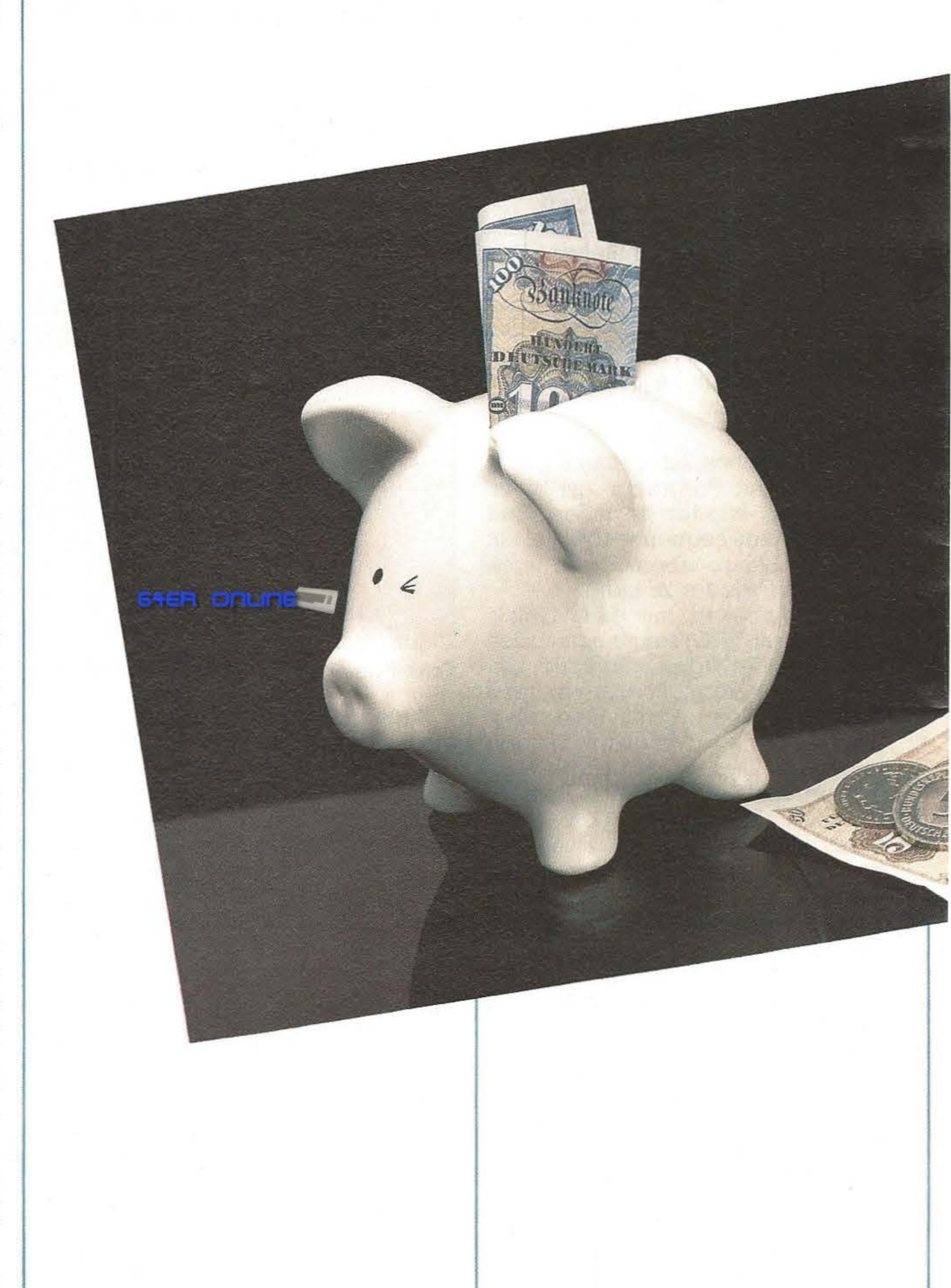
	Quick-Data-Drive	Turbo-Tape de luxe	Datasette	1541-Floppy
8 KByte Basic-Programm laden	6 sec.	23 sec.	2:52 min.	20 sec.
8 KByte Basic-Programm speichern	14 sec.	26 sec.	2:54 min.	25 sec.
2 KByte sequentielle Daten laden	37 sec.	—	1:54 min.	29 sec.
2 KByte sequentielle Daten speichern	54 sec.	—	1:56 min.	38 sec.

Tabelle. Die vier Kandidaten im Geschwindigkeitsvergleich.

Gut gekauft ist halb gespeichert

Die Diskette ist das wichtigste Speichermedium für Computerbesitzer. Unsere Marktübersicht hilft Ihnen, sich im »Diskettendschungel« zurechtzufinden.

Dem Disketten-Käufer wird die Bedeutung des Satzes »Qual der Wahl« recht bald bewußt. Disketten gibt es in den unterschiedlichsten Größen, und fast in jeder Farbe. Für den Commodore 64-Besitzer reduziert sich das Angebot auf 5¼-Zoll-Disketten. Die in diesem Format angebotenen Disketten sind allerdings nicht alle gleich. Sucht man die richtige Diskette, so ist es am besten einen Blick in das Laufwerk zu werfen. Das 1541-Laufwerk besitzt nur einen Schreib-/Lesekopf. Deshalb werden die Disketten auch nur auf einer Seite beschrieben. Unter dieser Voraussetzung eignen sich alle Disketten mit der Bezeichnung »SS« für Single Sided, die nur auf einer Seite geprüft sind. Disketten mit der Bezeichnung »DS« für Double Sided sind Garant für fehlerfreies Arbeiten auf beiden Seiten. Während SS-Disketten immer nur eine Schreibschutzkerbe besitzen, hat man bei DS die Möglichkeit zwischen Disketten mit einer oder zwei Kerben zu wählen. Natürlich kann man Disketten auch selbst mit einer zweiten Kerbe versehen, um dann auf die Rückseite zu schreiben — für Sicherheitskopien ein preiswerter Trick. Für viel benutzte Disketten eignet sich dies allerdings nicht, da der dem Schreib-/Lesekopf gegenüberliegende Andruckfilz immer über eine mit Daten beschriebene Seite schleift. Für die Güte einer Diskette gibt es weitere Kennbuchstaben. Die Buchstaben »SD« oder »1D« für Single Density (einfache Dichte) stehen meistens auf den etwas billigeren Disketten. Sie geben Aufschluß über die Dichte der Magnetisierungspartikel in der Diskettenbeschichtung. Finden Sie beispiels-





weise die Bezeichnung »DD« oder »2D« auf einer Diskette (Double Density = doppelte Dichte), besteht theoretisch die Möglichkeit, doppelt so viele Daten auf einer Diskette sicher abzulegen, als auf einer SD-Diskette. Für das 1541-Laufwerk genügt allerdings einfache Dichte. Nicht vernachlässigen dürfen Sie den Vermerk »Mit Verstärkungsring« (With Hub Ring) auf der Diskettenschachtel. Das heißt, daß um das Führungsloch der Diskette ein zweiter, verstärkend wirkender Ring geklebt ist. Da bei dem 1541-Laufwerk ein konischer Plastikring die Scheibe recht unsanft zentriert, ist es empfehlenswert, Disketten mit Verstärkungsring zu kaufen.

Dies alles sind Beurteilungskriterien, die Sie, ohne die Diskettenschachtel zu öffnen, feststellen können. Anders ist es aber mit den Qualitätsmerkmalen der Diskettenhülle, der Oberflächenstruktur der Magnetisierungsschicht und der Güte des Schutzvließes in der Diskette. Das ist für Sie in der Regel nicht so leicht überprüfbar. Die einzige Vergleichsmöglichkeit bieten die Datenblätter zu den einzelnen Disketten. Nur eines kann man hier mit Sicherheit sagen: Eine Spitzendiskette »schnurrt« leise vor sich hin ohne dabei die Tonlage nach oben oder unten zu verändern. Dies wäre nämlich ein Anzeichen dafür, daß sich die Diskette nicht einwandfrei in ihrem »Mantel« drehen kann. Das ist wichtig, weil es bei schwergängigen Disketten schnell zu Lesefehlern kommen kann.

Ein weiteres wichtiges Kriterium ist der Preis. Disketten werden im allgemeinen in 2er- oder in 10er-Packs angeboten. Hier lohnt es sich die Preise zu vergleichen und auf Sonderangebote zu achten. Beim Kauf größerer Mengen ist ein Preisnachlaß möglich. Um auf Stückzahlen dieser Größenordnung zu kommen, wendet man sich am besten an Clubs.

Wenn Sie nun alle diese Kriterien zu Ihrer Zufriedenheit abhaken konnten, werfen Sie noch einen Blick auf die Verpackung.

Um Ihnen die Entscheidung zu erleichtern, sind in der folgenden Marktübersicht die wichtigsten Argumente aufgezählt. Die Liste erhebt keinen Anspruch auf Vollständigkeit. Alle Daten beruhen auf Herstellerangaben.

(aw/og)

Marktübersicht 5¼″-Disketten

Hersteller/Anbieter Kriterium	Sektorierung: (h)ard, (s)oft	Seitenzahl/Dichte	Speicherkapazität in KByte	Anzahl lieferbarer Diskettenfarben, evtl. Aufpreis	10er Pack: a=Plastikbox, b=Kartonbox c=einfache Schachtel	Preisempfehlung: a=1 Diskette, b=10er Pack
3 M						
SS/DD	h,s	SS/DD	250	1	a,b	a 6,95
DS/DD	h,s	DS/DD	500	1	a,b	a 8,95
SS/DD 96 tpi	h,s	SS/DD	500	1	a,b	a 9,95
DS/DD 96 tpi	h,s	DS/HD	1000	1	a,b	a 11,30
DS/HD 96 tpi	s	DS/HD	1600	1	a	a 19,25
BASF						
FlexyDisk Science	h,s	SS/DD, DS/DD	1600	1	a,b	k.A.
FlexyDisk qualimetric	h,s	SS/SD, SS/DD	1600	1	a,b	k.A.
Comtec						
M11	h,s	SS/SD	125	6 ca. —,40	a,b	k.A.
M12	h,s	SS/DD	250	6 ca. —,40	a,b	k.A.
M22	h,s	DS/SS	500	6 ca. —,40	a,b	k.A.
M14	h,s	SS/DD	500	6 ca. —,40	a,b	k.A.
M24	h,s	DS/HD	1000	6 ca. —,40	a,b	.k.A.
Control Data						
1242-XX	h,s	SS/DD	250	1	a,b,c	b 57,50
1244-XX	h,s	DS/DD	500	1	a,b,c	b 69,50
1247-XX	h,s	SS/DD	500	1	a,b,c	b 74,—
1248-XX	h,s	DS/HD	1000	1	a,b,c	b 85,—
1249-XX	h,s	DS/HD	1600	1	a,b,c	b 129,—
Döbbelin & Boeder						
Disky 1	s	SS/SD	125	5 a. Anfr.	a,b,c	k.A.
Disky 1D	s	SS/SD	250	5 a. Anfr.	a,b,c	k.A.
Disky 2D	s	DS/DD	500	5 a. Anfr.	a,b,c	k.A.
Disky 1D96	s	SS/DD	500	5 a. Anfr.	a,b,c	k. A.
Disky 2D96	s	DS/DD	1000	5 a. Anfr.	a,b,c	k.A.
Disky two eye	s	DS/DD	500	5 a. Anfr.	a,b,c	k.A.
Disk top	s	DS/HD	1000	5 a. Anfr.	a,b,c	k.A.
Disky HD2	s	DS/HD	1600	5 a. Anfr.	a,b,c	k.A
Fuji						
MD1D	s	SS/DD	250	6 —,85	a,b	a 4,10
MD2D	s	DS/DD	500	6 —,90	a,b	a 5,95
MD2DD	s	DS/HD	1000	1	b	a 7,75
MD2FD	s	DS/HD	1600	1	b	a 9,95
Magna						
SS/DD 48 tpi	s	SS/DD	250	1	a	k.A.
DS/DD 48 tpi	s	DS/DD	500	1	a	k.A.
SS/DD 96 tpi	s	SS/DD	481	1	a	k.A.
DS/DD 86 tpi	s	DS/HD	962	1	a	k.A.

Hersteller/Anbieter Kriterium	Sektorierung: (h)ard, (s)oft	Seitenzahl/Dichte	Speicherkapazität in KByte	Anzahl lieferbarer Diskettenfarben, evtl. Aufpreis	10er Pack: a=Plastikbox, b=Kartonbox c=einfache Schachtel	Preisempfehlung: a=1 Diskette, b=10er Pack
Maxell						
MD1	s	SS/SD	125	1	a	a 7,50
MD1-D	s	SS/DD	250	1	a	a 7,85
MD2-D	s	DS/DD	500	1	a	a 11,70
MD1-DD	s	SS/DD	500	1	a	a 11,60
MD2-DD	s	DS/DD	500	1	a	a 15,35
MH1-10	s	SS/SD	250	1	a	a 7,50
MH1-16	s	SS/SD	250	1	a	a 7,50
MH2-10D	s	DS/DD	500	1	a	a 11,50
MH2-16D	h	DS/DD	500	1	a	a 11,50
MD2-256HD	s	DS/HD	1000	1	a	a 18,—
MD2-HD	s	DS/HD	1600	1	a	a 20,60
Memorex						
3203-5151	s	SS/SD	125	1	a	k.A.
3203-5251	h	SS/DD	250	1	a	k.A.
3203-5A51	s	SS/DD	250	1	a	k.A.
3203-5B51	s	SS/DD	250	1	a	k.A.
3203-5210	h	SS/DD	250	1	a	k.A.
3203-5A11	s	SS/DD	250	1	a	k.A.
3203-5B11	s	SS/DD	250	1	a	k.A.
3203-5261	h	DS/DD	500	1	a	k.A.
3203-5A61	s	DS/DD	500	1	a	k.A.
3203-5B62	s	DS/DD	500	1	a	k.A.
3203-5220	h	DS/DD	500	1	a	k.A.
3203-5A21	s	DS/DD	500	1	a	k.A.
3203-5B21	s	DS/DD	500	1	a	k.A.
3203-5410	h	SS/DD	500	1	a	k.A.
3203-5C11	s	SS/DD	500	1	a	k.A.
3203-5D11	s	SS/DD	500	1	a	k.A.
3203-5420	h	SS/SD	1000	1	a	k.A.
3203-5C21	s	DS/HD	1000	1	a	k.A.
3203-5D21	s	DS/HD	1000	1	a	k.A.
3203-5660	h	DS/HD	1600	1	a	k.A.
Marcom						
Elephant EMS 2	h	SS/DD	250	1	c	b 55,—
Elephant EMS 3	s	SS/DD	250	1	c	b 55,—
Elephant EMS 4	s	SS/DD	250	1	c	b 55,—
Elephant EMS 5	h	DS/DD	500	1	c	b 70,—
Elephant EMS 6	s	DS/DD	500	1	c	b 70,—
Elephant EMS 7	s	DS/DD	500	1	c	b 70,—
Elephant EMS 8	h	SS/DD	500	1	c	b 75,—
Elephant EMS 9	s	SS/DD	500	1	c	b 75,—
Elephant EMS 10	h	DSS/HD	1000	1	c	b 85,—
Elephant EMS 11	s	DS/HD	1000	1	c	b 85,—
Nashua						
MD1	h,s	SS/SD	125	1	a	k.A.
MD1D	h,s	SS/DD	250	1	a	k.A.
MD2D	h,s	DS/DD	500	1	a+b	k.A.
MD1F	h,s	SS/DD	500	1	a+b	k.A.
MD2F	h,s	DS/HD	1000	1	a+b	k.A.
Panasonic						
EBF MD1D	s	SS/DD	164	1	b	a 4,80
EBF MD2D	s	DS/DD	328	1	b	a 7,15
EBF MD2DD	s	DS/DD	655	1	b	a 10,20
Pelikan						
MD1 D	s	SS/DD	250	1	b	k.A.
MD2D	s	DS/DD	500	1	b	k.A.
MD1DD	s	SS/DD	500	1	b	k.A.
MD2DD	s	DS/HD	1000	1	b	k.A.
MD2HD	s	DS/HD	1600	1	b	k.A.
Perfectdata International						
1S2D	s	SS/DD	500	2	a	a 2,90
2S2D	s	DS/DD	500	2	a	a 3,50
1S2D	s	SS/HD	1000	2	a	a 4,—
2S2D	s	DS/HD	1000	2	a	a 4,50
Rhone Poulenc						
MN1DD	h,s	SS/DD	250	1	a,b,c	k.A.
MN2DD	h,s	DS/DD	500	1	a,b,c	k.A.
MN2DD-2	h,s	DS/HD	1000	1	a,b,c	k.A.
MN2HD	h,s	DS/HD	1600	1	a,b,c	k.A.
Sentinel						
5H-11 5S-11	h,s	SS/SD	125	6	a,b,c	k.A.
5H-12 5S-12	h,s	SS/DD	250	6	a,b,c	k.A.
5H-22 5S-22	h,s	DS/DD	500	6	a,b,c	k.A.
5S-14 5S-14	h,s	SS/DD	500	6	a,b,c	k.A.
5H-24 5S-24	h,s	SD/HD	1000	6	a,b,c	k.A.
5S-28	h,s	DS/HD	1600	6	a,b,c	k.A.
Verbatim						
Verex MD 150-01	s	SS/SD	125	1	c	b 35,—
Verex MD 200-01	s	SS/DD	250	1	c	b 39,—
Verex MD [illegible]-01	s	DS/DD	500	1	c	b 49,—
Datalife MD 525-01	s	SS/DD	250	1	c	b 49,—
Datalife MD 550-01	s	DS/DD	500	1	c	b 69,50
Datalife MD 577-01	s	DS/DD	500	1	c	b 69,—
Datalife MD 557-01	s	DS/HD	1000	1	c	b 89,—
Datalife MD HD	s	DS/HD	1600	1	c	b 120,—
Vobis						
Multilife 1D1S	s	SS/SD	125	1	b	b 29,—
Multilife 2D1S	s	SS/DD	250	1	b	b 35,—
Multilife 1D2S	s	DS/SD	250	1	b	b 39,—
Multilife 1D2S	s	DS/SD	250	1	b	b 49,—
Wende-disketten						
Multilife 2D2S	s	DS/DD	500	6	b	b 69,—
Wabash DataTech						
M11	s	SS/SD	125	1	b	k.A.
M13	h,s	SS/DD	250	1	b	k.A.
M14	h,s	DS/DD	500	1	b	k.A.
M16	h,s	DS/HD	1000	1	b	k.A.
M1B	s	DS/HD	1600	1	b	k.A.
Xidex						
5012-1000 5112-1000 5212-1000	h,s	SS/DD	250	1	a,b	k.A.
5022-1000 5122-1000 5222-1000	h,s	DS/DD	500	1	a,b	k.A.
5012-2000 5122-2000 5212-2000	h,s	SS/DD	500	1	a,b	k.A.
5022-2000 5122-2000 5222-2000	h,s	DS/HD	1000	1	a,b	k.A.

EPROM-Trans — Die Super-Erweiterung

Der ROM-Speicher des C 64 ließ sich bislang nur mit Steckplatinen im Expansion-Port vergrößern. Mit einem raffinierten Trick geht es auch intern.



Bild 6. So wird EPROM-Trans in den C 64 eingebaut

Wen hat es nicht schon geärgert, daß die Grundversion des C 64 kein Monitorprogramm aufweist, das nach dem Einschalten des Computers sofort zur Verfügung steht? Deshalb wurde eine Möglichkeit gesucht (und gefunden), mit der man Maschinenprogramme von einem festinstalliertem EPROM in den Bereich $C000 bis $CFFF transformieren kann, ohne für das EPROM vorhandenen Speicherplatz in Anspruch zu nehmen.

Um diese Schaltung zu realisieren, wurde bei den C 64-Entwicklern einiges abgeguckt. Es soll hier nicht die Funktionsweise des C 64 im Detail erläutert werden (das würde zu weit führen), doch läßt es sich für das Verständnis dieser Schaltung nicht vermeiden, auf den Speicherbereich $A000 bis $BFFF (hier liegt das Basic-ROM) näher einzugehen.

Wird dieser Bereich angesprochen, so wird das Basic-ROM über das sogenannte PLA-Chip (Programable-Logic-Array), auch Adreßraummanager genannt, unter bestimmten Voraussetzungen (zum Beispiel Lesen) aktiviert, das heißt der $\overline{CE}$ (Chip-Enable)-Anschluß auf Low-Potential gelegt. Drückt man zum Beispiel »RETURN«, so wird zur Adresse $A560 gesprungen. Diese liegt im Speicherbereich, in dem das Basic-ROM liegt. Dieses ROM ist folglich aktiviert, und das dort stehende Maschinenprogramm (Eingabe einer Zeile) wird abgearbeitet.

Wie aus Bild 1 ersichtlich, wurde dieser Bereich ($A000 bis $BFFF) benutzt. Über diesem ROM-Bereich liegt zum einen das »EPROM-Trans« ($A000 bis $AFFF) und zum anderen die EPROMs 1 bis 3 ($B000 bis $BFFF). In letzteren steht jeweils ein beliebiges Programm (bis 4 KByte), welches in den Bereich $C000 bis $CFFF transformiert werden soll. Ich habe zum Beispiel in EPROM 1 den »SMON« und in EPROM 2 »Fast Tape« abgelegt, EPROM 3 ist noch unbenutzt. Das Maschinenprogramm in EPROM-Trans (siehe Listing) soll den Inhalt des jeweils selektierten EPROMs 1 bis 3 von dem Bereich $B000 bis $BFFF in den Bereich $C000 bis $CFFF transformieren.

Wie kann dies jedoch funktionieren, wenn in diesem Bereich wie oben erwähnt, beim Lesen immer das Basic-ROM aktiviert ist? Für die Speichertransformation wird der Inhalt des Basic-ROMs nicht benötigt. Drückt man nun einen der drei Taster (Bild 2), wird die $\overline{CE}$-Leitung des Basic-ROMs auf die Zusatzschaltung umgeleitet. Ist folglich ein Taster gedrückt, dann ist das Basic-ROM nicht mehr ansprechbar; kein Taster gedrückt entspricht dem Normalzustand des Computers.

Wie aus dem Listing ersichtlich, liegt bei der Adresse $0560 (≙ $A560) des Maschinenprogramms beim EPROM-Trans ein Sprungbefehl zur Adresse $A000, wo dann die Speichertransformation startet und mit einem Sprungbefehl nach $C000 endet. Will man diese Speichertransformation starten, muß der Computer zur Adresse $A560 oder $A000 springen. Wie oben schon erwähnt, wird die Adresse $A560 automatisch beim Drücken der »RETURN«-Taste angesprungen. Wird zum Beispiel EPROM 1 durch Betätigen von Taster 1 selektiert und dann »RETURN« gedrückt, startet das Maschinenprogramm in EPROM-Trans. Nun wird mit Adreßbit A12 zwischen den Bereichen $Axxx und $Bxxx hin- und hergeschaltet und so der komplette Inhalt von EPROM 1 nach $C000 bis $CFFF kopiert. Am Ende der Transformation wird das Programm durch den JMP-Befehl gestartet, so daß sich zum Beispiel der SMON gleich mit seiner Registeranzeige meldet. Nach Loslassen des Tasters steht auch das Basic-ROM wieder zur Verfügung.

Ein eventuell vorhandenes Basic-Programm wird während der Speichertransformation selbstverständlich nicht beschädigt. Änderungen sind nur im Bereich $C000 bis $CFFF zu registrieren. Die Transformation weist keine Zeitprobleme auf; wenn man »RETURN« betätigt (≙ Start der Transformation), meldet sich auch gleich der SMON (um beim Beispiel zu bleiben).

In EPROM 1 bis 3 wird das Maschinenprogramm so abgelegt, wie es auch in $C000 bis $CFFF stehen würde, da ja $B000 nach $C000, $B001 nach $C001 und $BFFF nach $CFFF transformiert wird.

In Bild 2 ist die Steuerung der $\overline{CE}$-Eingänge der einzelnen EPROMs und des Basic-ROMs dargestellt.

Das IC 74LS139 besteht aus zwei 1 auf 4 Demultiplexer DM1 und DM2.

Je nach Codierung der Eingangsadresse (A0 und A1) wird der Eingang E auf den entsprechenden Ausgang D0 bis D3 gelegt (Tabelle 1), ansonsten haben die Ausgänge High-Potential.

Realisierung der Hardware

Mit DM1 erfolgt die eigentliche Selektierung der EPROMs. Es bestehen vier Möglichkeiten: Basic-ROM und EPROM 1 bis 3. Der Adreßcode ist abhängig von den Schaltzuständen der Taster T1 bis T3. Aktiviert werden alle EPROMs mit »Active Low« (Low-Potential). $\overline{\text{BASIC}}$ ist die Leitung, die normalerweise das Basic-ROM aktiviert, also dann Low-Pegel hat, wenn im Bereich $A000 bis $BFFF gelesen wird. Dieses Signal ist an die Eingänge E der Demultiplexer DM1 und DM2 geführt. Liegt $\overline{\text{BASIC}}$ auf, kann kein Baustein aktiviert werden, denn die Ausgänge D0 bis D3 von DM1 und DM2 sind high oder entsprechend dem Pegel an E (hier dann auch high). Am $\overline{\text{CE}}$-Eingang vom Basic-ROM steht folglich ein High-Pegel, entsprechendes gilt für EPROM-Trans. Die Ausgänge D1 bis D3 von DM1 werden invertiert, so daß an den betreffenden Eingängen der NAND-Gatter Low-Potential liegt. Bekanntlich ist aber der Ausgang eines NAND-Gatters nur dann low, wenn beide Eingänge high sind. Folglich werden EPROM 1 bis 3 nicht aktiviert. Kommen wir jetzt zum interessanteren Teil: $\overline{\text{Basic}}$ für LOW. Als erstes wollen wir prüfen, ob der Computer »normal« arbeitet, wenn keine Taste gedrückt ist:

a) keine Taste gedrückt:

Wenn $\overline{\text{BASIC}}$ low ist, muß in diesem Fall das Basic-ROM aktiviert werden. Durch die Tasterstellungen liegt an den Eingängen A0, A1 von DM1 Low-Potential. Dies bedeutet, die Information an E ($\overline{\text{BASIC}}$ = low) wird zum Ausgang D0 (vergleiche Tabelle 1) übertragen. Das Basic-ROM wird also aktiviert (unabhängig von A12). EPROM 1 bis 3 können nicht aktiviert werden, da die Ausgänge D1 bis D3 von DM1 High-Potential führen (siehe oben). Aber was ist mit EPROM-Trans? Hier (DM2) wird das Eingangssignal E auf den Ausgang D2 übertragen, wenn A0 low und A1 high ist. A1 ist aber abhängig vom $\overline{\text{CE}}$-Signal des Basic-ROMs. Ist dieses aktiviert (low), kann EPROM-Trans nicht aktiviert werden. Der Computer arbeitet in diesem Fall also völlig normal.

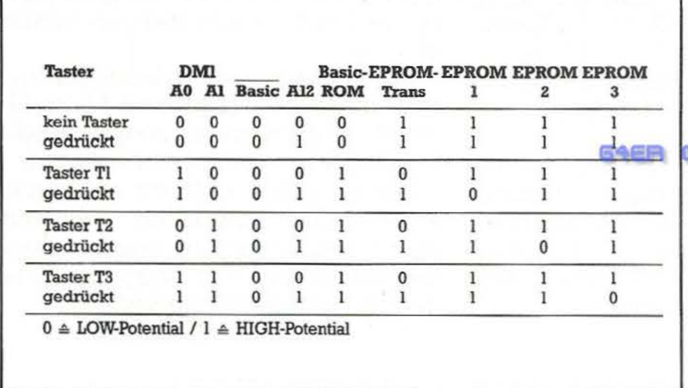
b) Taster T1 wird gedrückt

Wird Taster T1 gedrückt, liegt an A0 von DM1 über den Pull-up-Widerstand (1,5 kOhm) high, A1 bleibt über T2 und T3 auf low. Diese Adreßcodierung bedeutet, daß die Information an E auf den Ausgang D1 übertragen wird. Durch den Inverter liegt also am NAND-Gatter (Pin 1) high.

Als Adresse an DM2 liegt an A1 jetzt high, da das Basic-ROM nicht aktiviert wird. A0 ist abhängig vom Adreßbit A12. Wie erwähnt ist A12 für den Bereich $Axxx low. Die Eingangsinformation (low) wird auf D2 gelegt und EPROM-Trans wird aktiviert. An Pin 2 vom NAND-Gatter liegt Low-Potential (high von D3 invertiert), weshalb EPROM 1 nicht aktiviert wird. Wechselt Adreßbit A12 seinen Zustand (von $Axxx → $Bxxx) wird Eingang E von DM2 mit Ausgang D3 zusammengeschaltet, denn A0=0 und A1=1. EPROM-Trans wird nicht mehr aktiviert (D2 → High). An D3 liegt nun aber Low-Potential welches über einen Inverter auf das NAND-Gatter (Pin 2) geführt wird. An Pin 1 und 2 dieses Gatters liegt folglich high und am Ausgang liegt Low-Pegel. EPROM 1 wird aktiviert.

c) Taster T2/T3 wird gedrückt

Das Prinzip ist das gleiche wie eben beschrieben. Es ändert sich nur die Adreßcodierung von DM1.



Taster	DM1 A0	DM1 A1	$\overline{\text{Basic}}$	A12	Basic-ROM	EPROM-Trans	EPROM 1	EPROM 2	EPROM 3
kein Taster	0	0	0	0	0	1	1	1	1
gedrückt	0	0	0	1	0	1	1	1	1
Taster T1	1	0	0	0	1	0	1	1	1
gedrückt	1	0	0	1	1	1	0	1	1
Taster T2	0	1	0	0	1	0	1	1	1
gedrückt	0	1	0	1	1	1	1	0	1
Taster T3	1	1	0	0	1	0	1	1	1
gedrückt	1	1	0	1	1	1	1	1	0

0 ≙ LOW-Potential / 1 ≙ HIGH-Potential

Tabelle 1. Codierung des Demultiplexers (H=High)

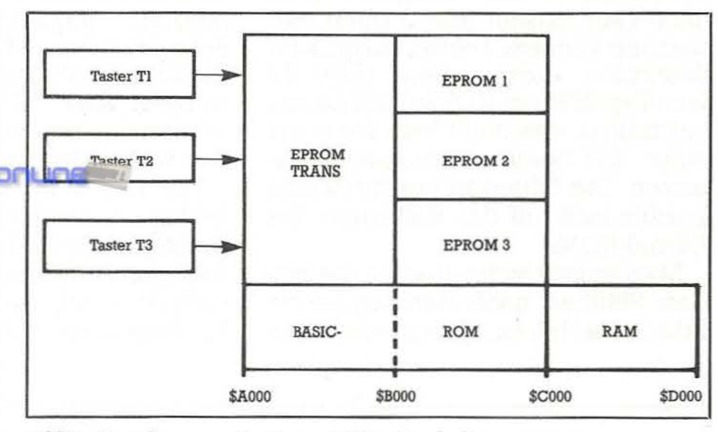


Bild 1. Speicherorganisation mit Zusatzschaltung

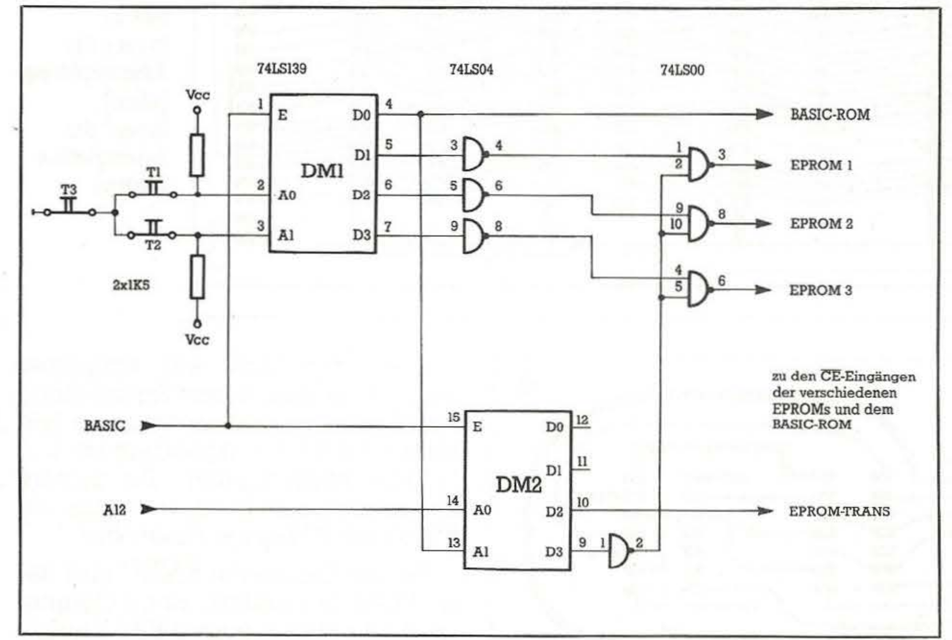


Bild 2. Zusatzschaltung zur Steuerung der verschiedenen EPROMs und dem Basic-ROM

Um ein Kabelgewirr zu vermeiden »zapft« man am besten ein ROM im Computer an, das heißt man erstellt eine Adapterplatine, auf der sämtliche zusätzliche EPROMs sowie das entwendete ROM Platz finden und Adreß- und Datenbus parallel verbunden werden. Diese Platine wird dann in den Steckplatz des herausgenommenen ROMs wie ein einzelnes IC gesteckt, wobei die Anschlüsse des ROMs direkt verbun-

Kernal oder Basic-ROM?

den sind, während die zusätzlichen Anschlüsse der EPROMs (±, $\overline{\text{CE}}$, $\overline{\text{OE}}$, ...) extern herausgeführt werden. Welches ROM man verwendet, hängt von den ganz individuellen Wünschen ab. Ich habe mich für das Kernal entschieden, da ich so für ein geändertes Betriebssystem gleich einen EPROM-Steckplatz mit vorgesehen habe, so daß zum Beispiel diejenigen, die eine Betriebssystem-Erweiterung benutzen, ohne Änderungen die Adapterplatine nach dem Layout Bild 3 oben verwenden können. Dieser zusätzliche Steckplatz »Kernal neu« (EPROM vom Typ 2764) ist 100%ig zum Kernal verdrahtet, das heißt man kann nur einen der beiden Kernalplätze besetzen. Die folgende Beschreibung bezieht sich auf das Entfernen des Kernal-ROMs.

Man erstellt sich zunächst die beiden Platinen nach den Layouts in Bild 3. Nachdem die angedeuteten Löcher für Bauteile und Drahtbrücken gebohrt (1 mm) sind, beginnt das Bestücken, wobei es ratsam ist, mit den von den Abmessungen niedrigsten Bauteilen zu beginnen. So ergibt sich folgende Reihenfolge: Drahtbrücken, Widerstände, Fassungen. Der Adaptersockel wird unter die IC-Fassung des auf dieser Platine vorgesehenen Platzes für das Kernal festgelötet, da er ja quasi nur eine Verlängerung der Kernal-Pins ist. Dies muß so durchgeführt werden, daß der Adaptersockel möglichst rechtwinklig zur Platine steht und in eine 24polige IC-Fassung paßt. Am einfachsten kann dieses realisiert werden, wenn man den Adaptersockel in eine 24polige Fassung steckt, und dann anlötet und ausrichtet.

Zwischen Zusatzplatine und Adapterplatine werden sieben Verbindungskabel gezogen, wobei die zugehörigen Anschlußpunkte den Bestückungsplänen Bild 4 zu entnehmen sind. Auf der Adapterplatine kann man hierbei eine Steckverbindung (10polig) vorsehen (zum Beispiel: Typ MS-25-10; Völkner-Elektronik, Postfach 5330, 33 Braunschweig; Preis 3,25 Mark), damit die beiden Platinen nicht starr miteinander verbunden sind.

Die Taster T1 bis T3 werden als letztes — eventuell erst nach dem Einbau — nach Bild 4 unten angeschlossen. Die Qualität der Taster ist unbedeutend, da der Faktor des Prellens keine Rolle spielt.

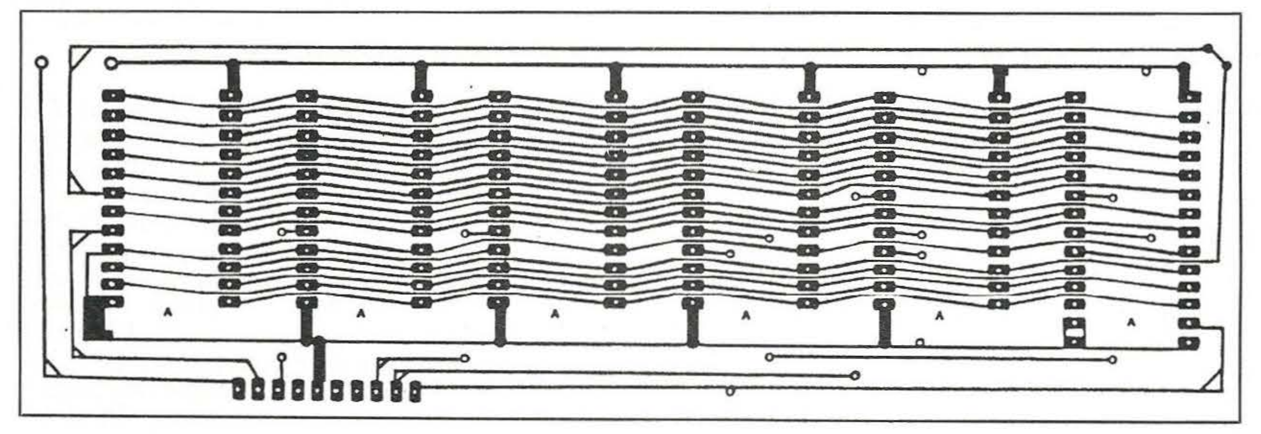
Verwendet man das hier abgebildete Platinenlayout, so müssen EPROM 1 bis 3 vom Typ 2532 sein, EPROM-Trans vom Typ 2716. Hier sollten wir noch einmal einen Blick auf Bild 5 werfen: Bis auf die $\overline{\text{CE}}$-Anschlüsse, die zur Zusatzplatine geführt werden, sind die Leitungen fest mit den des Kernal-ROMs verdrahtet (also Adreß- und Datenbus). Die restlichen epromspezifischen Anschlüsse ($\overline{\text{OE}}$, V_{pp}, $\overline{\text{PRG}}$) werden je nach Typ für EPROM-Lesen fest auf low oder high gelegt. Die Stromversorgung liegt beim Kernal an den Pins 12 (Masse) und 24 (V_{cc}= 5V); A12 entspricht Pin 21.

Einbau in den Computer

Zum Öffnen des Computers (vorher sämtliche Peripherie abziehen inklusive Stromversorgung, Achtung Garantieverlust) müssen die drei Schrauben an der vorderen Unterseite gelöst werden. Hinten sind Ober- und Unterteil des Gehäuses nur zusammengesteckt, so daß durch Anheben des Oberteils sich beide Gehäuseteile trennen lassen (Bild 6).

Jetzt müssen die beiden Steckverbindungen für Tastatur und Leuchtdiode gelöst werden, so daß beide Gehäuseteile völlig getrennt sind.

Als nächstes entfernt man das Kernal-ROM (Steckplatz U4) und steckt dieses in den entsprechenden Sockel auf der Adapterplatine gemäß dem Bestückungsplan.



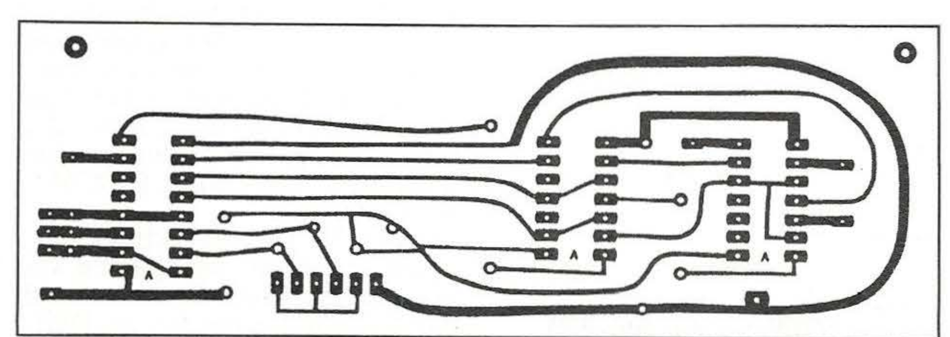


Bild 3. Layout der Adapterplatine (oben); Layout der Zusatzplatine (unten)

Sollte das ROM fest eingelötet sein, so ist dies unter Verwendung von Entlötlitze und einer guten Lötstation (nicht mit irgendeinem Lötkolben) herauszulöten. An diesem freigewordenen Platz wird dann ein 24poliger IC-Sockel eingesetzt.

Um die Leitungen $\overline{\text{BASIC}}$ und Basic-ROM zu erhalten, ist im Computer die Leiterbahn nach Bild 7 aufzutrennen. Hierzu nimmt man einen

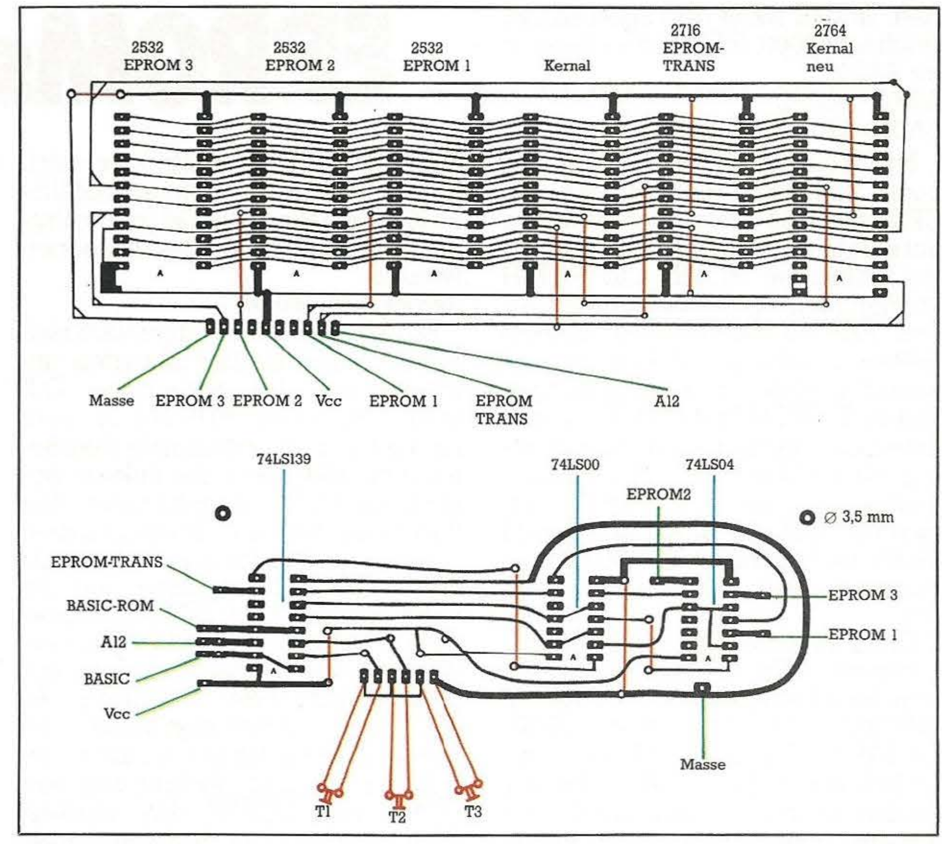


Bild 4. Bestückungspläne: Adapterplatine (oben); Zusatzplatine (unten). Die Kerben auf den ICs zeigen immer nach unten. Abgebildet ist die jeweilige Lötseite der Platinen.

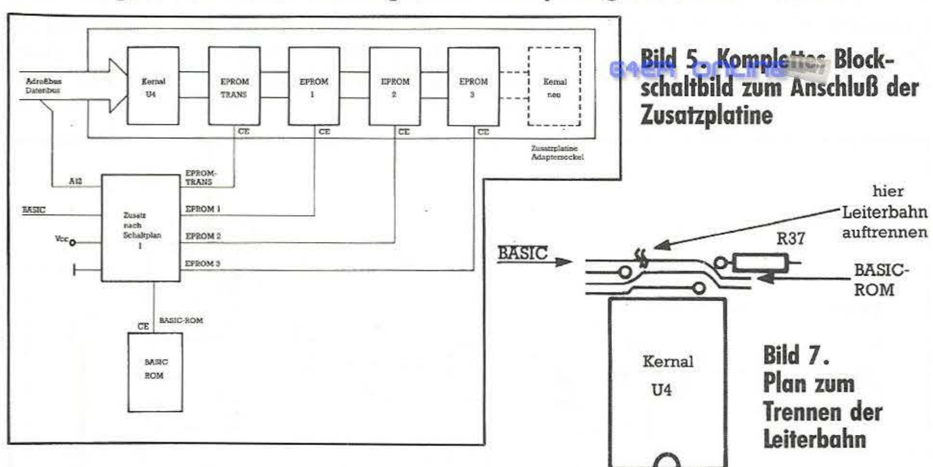


Bild 5. Komplettes Blockschaltbild zum Anschluß der Zusatzplatine

hier Leiterbahn auftrennen

R37

BASIC

BASIC-ROM

Kernal U4

Bild 7. Plan zum Trennen der Leiterbahn

kleinen Schraubenzieher und unterbricht die Leiterbahn durch vorsichtiges Kratzen. Die grüne Schutzschicht auf der Leiterbahn wird an beiden Seiten (rechts und links von der Unterbrechung) entfernt, so daß das Kupfer sichtbar wird. Nun wird an diesen Stellen jeweils ein flexibles Kabel mit etwas Fingerspitzengefühl angelötet. Die eine Leitung entspricht dann $\overline{\text{BASIC}}$ (vom PLA), die andere Basic-ROM (zu $\overline{\text{CE}}$ dieses ROMs) und sind als solche bezeichnet auf der Zusatzplatine zu finden und entsprechend zu verdrahten (eventuell mit Steckverbindung auf der Zusatzplatine).
Diejenigen, die anstatt des Kernals das Basic-ROM anzapfen

Fortsetzung auf Seite 140

```
10 FOR I=32768 TO 32947                        <213>
20 READ X:POKE I,X: S=S+X: NEXT                <222>
30 DATA 162,  0,189,  0,176,157,  0,192,23
   2,208,247,162,  0,189,  0,177,157           <180>
40 DATA   0,193,232,208,247,162,  0,189,
   0,178,157,  0,194,232,208,247,162           <099>
50 DATA   0,189,  0,179,157,  0,195,232,20
   8,247,162,  0,189,  0,180,157,  0           <154>
60 DATA 196,232,208,247,162,  0,189,  0,18
   1,157,  0,197,232,208,247,162,  0           <102>
70 DATA 189,  0,182,157,  0,198,232,208,24
   7,162,  0,189,  0,183,157,  0,199           <185>
80 DATA 232,208,247,162,  0,189,  0,184,15
   7,  0,200,232,208,247,162,  0,189           <058>
90 DATA   0,185,157,  0,201,232,208,247,16
   2,  0,189,  0,186,157,  0,202,232           <153>
100 DATA 208,247,162,  0,189,  0,187,157,
    0,203,232,208,247,162,  0,189,  0          <137>
110 DATA 188,157,  0,204,232,208,247,162,
    0,189,  0,189,157,  0,205,232,208          <013>
120 DATA 247,162,  0,189,  0,190,157,  0,2
    06,232,208,247,162,  0,189,  0,191         <149>
130 DATA 157,  0,207,232,208,247, 76,  0,1
    92,247                                     <213>
140 IF S<>    25763 THEN PRINT "DATA-FEHLE
    R !": END                                  <232>
150 POKE 34144,76:POKE 34145,0:POKE 34146,
    160                                        <229>
160 PRINT "O.K."                               <219>
```

Listing 2. Basic-Lader des Programms

```
,A000  A2 00     LDX #00      Block 1
,A002  BD 00 B0  LDA B000,X
,A005  9D 00 C0  STA C000,X
,A008  E8        INX
,A009  D0 F7     BNE A002
,A00B  A2 00     LDX #00      Block 2
,A00D  BD 00 B1  LDA B100,X
,A010  9D 00 C1  STA C100,X
,A013  E8        INX
,A014  D0 F7     BNE A00D
,A016  A2 00     LDX #00      Block 3
,A018  BD 00 B2  LDA B200,X
,A01B  9D 00 C2  STA C200,X
,A01E  E8        INX
,A01F  D0 F7     BNE A018
,A021  A2 00     LDX #00      Block 4
,A023  BD 00 B3  LDA B300,X
,A026  9D 00 C3  STA C300,X
,A029  E8        INX
,A02A  D0 F7     BNE A023
,A02C  A2 00     LDX #00      Block 5
,A02E  BD 00 B4  LDA B400,X
,A031  9D 00 C4  STA C400,X
,A034  E8        INX
,A035  D0 F7     BNE A02E
,A037  A2 00     LDX #00      Block 6
,A039  BD 00 B5  LDA B500,X
,A03C  9D 00 C5  STA C500,X
,A03F  E8        INX
,A040  D0 F7     BNE A039
,A042  A2 00     LDX #00      Block 7
,A044  BD 00 B6  LDA B600,X
,A047  9D 00 C6  STA C600,X
,A04A  E8        INX
,A04B  D0 F7     BNE A044
,A04D  A2 00     LDX #00      Block 8
,A04F  BD 00 B7  LDA B700,X
,A052  9D 00 C7  STA C700,X
,A055  E8        INX
,A056  D0 F7     BNE A04F
,A058  A2 00     LDX #00      Block 9
,A05A  BD 00 B8  LDA B800,X
,A05D  9D 00 C8  STA C800,X
,A060  E8        INX
,A061  D0 F7     BNE A05A
,A063  A2 00     LDX #00      Block 10
,A065  BD 00 B9  LDA B900,X
,A068  9D 00 C9  STA C900,X
,A06B  E8        INX
,A06C  D0 F7     BNE A065
,A06E  A2 00     LDX #00      Block 11
,A070  BD 00 BA  LDA BA00,X
,A073  9D 00 CA  STA CA00,X
,A076  E8        INX
,A077  D0 F7     BNE A070
,A079  A2 00     LDX #00      Block 12
,A07B  BD 00 BB  LDA BB00,X
,A07E  9D 00 CB  STA CB00,X
,A081  E8        INX
,A082  D0 F7     BNE A07B
,A084  A2 00     LDX #00      Block 13
,A086  BD 00 BC  LDA BC00,X
,A089  9D 00 CC  STA CC00,X
,A08C  E8        INX
,A08D  D0 F7     BNE A086
,A08F  A2 00     LDX #00      Block 14
,A091  BD 00 BD  LDA BD00,X
,A094  9D 00 CD  STA CD00,X
,A097  E8        INX
,A098  D0 F7     BNE A091
,A09A  A2 00     LDX #00      Block 15
,A09C  BD 00 BE  LDA BE00,X
,A09F  9D 00 CE  STA CE00,X
,A0A2  E8        INX
,A0A3  D0 F7     BNE A09C
,A0A5  A2 00     LDX #00      Block 16
,A0A7  BD 00 BF  LDA BF00,X
,A0AA  9D 00 CF  STA CF00,X
,A0AD  E8        INX
,A0AE  D0 F7     BNE A0A7
,A0B0  4C 00 C0  JMP C000
--------------------------------------------
 :
 :                             Sprung: Start
,A560  4C 00 A0  JMP A000
--------------------------------------------
```

Listing 1. Speichertransformation $Bxxx nach $Cxxx.

EPROMs

Das Laden von Programmen dauert, selbst wenn man Hypra-Load verwendet, doch immer noch eine geraume Zeit. Viel praktischer sind die Ihnen sicher schon bekannten EPROMs. Der Anschluß von EPROMs stellt beim C 64 kein allzu großes Problem dar. Am Expansion-Port befinden sich alle für diesen Zweck wichtigen Adreß-, Daten- und Steuerleitungen. Analog dazu verfügt auch ein EPROM über Adreß-, Daten- und einige Steueranschlüsse. Letztere unterteilen sich in Pins für die Stromversorgung (0 V, +5 V), Pins zur Programmierung und Pins zur letztendlich gewünschten Verbindung mit dem Bus-System des C 64. Da man nach einer Programmierung des EPROMs die Anschlüsse für die Programmiersteuerung (PGM) nicht mehr benötigt, diese aber im Betrieb einen definierten Zustand haben müssen, werden beide auf logisch High (+5V) gelegt. Der Anschluß »Chip Enable« (CE = Chipauswahl) teilt dem EPROM mit, daß es angesprochen ist, und »Output Enable« (OE = Ausgabe möglich) fordert das EPROM auf, die Daten an den Datenbus zu legen. Danach stehen die Daten genauso zur Verfügung, als ob der Computer sie aus dem RAM lesen würde.

Zwei Platinen zur Wahl

Zum Aufbau der Schaltung ist es am besten, eine Platine anzufertigen, die in den Expansion-Port paßt. Dabei haben wir uns für zwei verschiedene Aufführungen, eine mit 8 KByte für kleinere Programme und eine mit 16 KByte Speicherplatz entschieden. Es können wahlweise EPROMs vom Typ 2732 (4KByte) oder 2764 (8 KByte), verwendet werden. Zusätzlich bietet die 16 KByte-Platine mehrere Betriebsmodi, die im einzelnen noch beschrieben werden. Doch zurück zum Anschluß der EPROMs an den C 64. Da ein EPROM maximal 8 KByte Speicherplatz hat und damit sicherlich nicht den ganzen Adreßraum des 6510-Prozessors ausfüllt, muß der C 64 den Speicherbereich für das EPROM festlegen. Für diesen Zweck sind zwei Leitungen vorgesehen: ROML und ROMH. Beide sind »aktiv-low«, das heißt sie wirken erst, wenn sie vom Zustand »1« in den Zustand »0« übergehen, beziehungsweise wenn sie ein Potential von 0 Volt gegenüber Masse haben. Diese Leitungen werden mit den Anschlüssen CE und OE (beide ebenfalls aktiv-low) des EPROMs verbunden. ROML wählt den Speicherbereich ab $8000, ROMH den Bereich ab $A000:

$8000: ROML ⨯ OE
$A000: ROMH ⨯ CE

Jetzt sollte man dem Computer noch mitteilen, daß er auf die EPROMs und nicht, wie gewohnt, auf sein internes RAM zugreift, denn die Leitungen ROML und ROMH können nicht direkt beeinflußt werden. Erst die Verwendung von zwei weiteren Leitungen bringt das gewünschte Ergebis. Diese Leitungen heißen EXROM und GAME. Sobald eine dieser Leitungen auf Masse gelegt wird, teilt sie der PLA (Programmable Logic Array = Adreßmanager) mit, daß die Daten im Bereich $8000, beziehungsweise $A000 vom Expansion-Port geholt werden sollen.

Damit sind die wesentlichen Funktionen der 8-KByte-Platine beschrieben. Sie ist so konzipiert, daß sie ein EPROM in den Bereich von $8000 bis $9FFF einblendet. Mit den vier Lötbrücken auf der Oberseite der Platine kann allerdings auch eine andere Speicheraufteilung eingestellt werden. Bei Bedarf trennen sie die beiden geschlossenen Lötbrücken auf und verbinden die beiden geöffneten. Damit wird der Speicherbereich von $A000 bis $BFFF für die EPROMs selektiert. Dies ist jedoch eine sehr ausgefallene Konfiguration, die Sie in der Praxis wahrscheinlich selten brauchen werden.

Mehrere Betriebsarten

Bei der 16-KByte-Platine wird zwischen mehreren Betriebsarten unterschieden. Im einfachsten Fall wird das rechte EPROM A (vom Expansion-Port gesehen) in den Bereich ab $8000 und das linke in Bereich ab $A000 eingeblendet. Aus den einleitenden Betrachtungen wissen wir, daß für den Bereich ab $8000 EXROM auf »Low« und für den Bereich »$A000 GAME auf »low« gelegt (mit der Masse Leitung verbunden) werden müssen. Dies geschieht mit den Schaltern S4 (EXROM) beziehungsweise S3 (GAME). Weiterhin erfolgt über den Schalter S5 eine Verbindung von ROML mit CE/OE des rechten EPROMs und über den Schalter S1 ein Kontakt zwischen ROMH und CE/OE des linken EPROMs.

Als nächstes soll die Schaltung so verändert werden, daß beide EPROMs im Bereich ab $8000 ste-

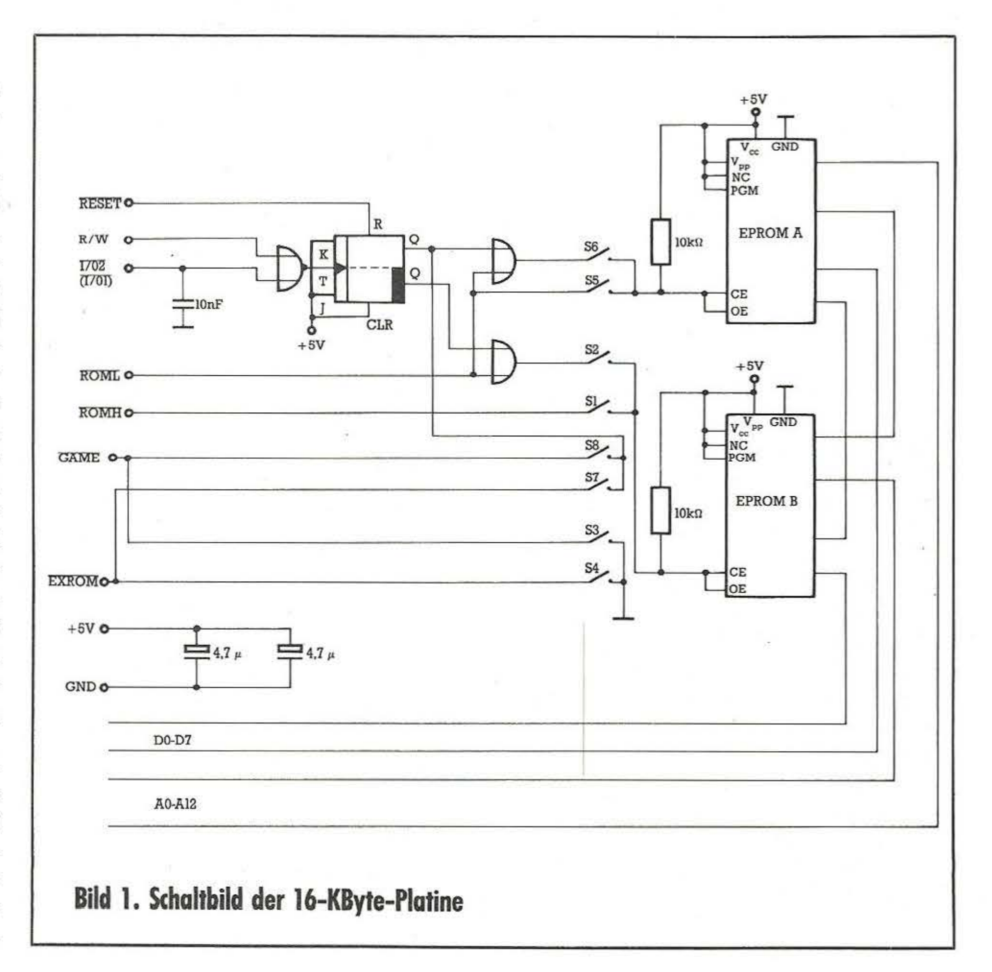


Bild 1. Schaltbild der 16-KByte-Platine

im Expansion-Port

Es gibt kaum ein Speichermedium, von dem Daten so schnell geladen werden können wie von EPROMs. Bauen Sie sich selbst eine Erweiterungsplatine mit wahlweise 8 oder 16 KByte Speicher!

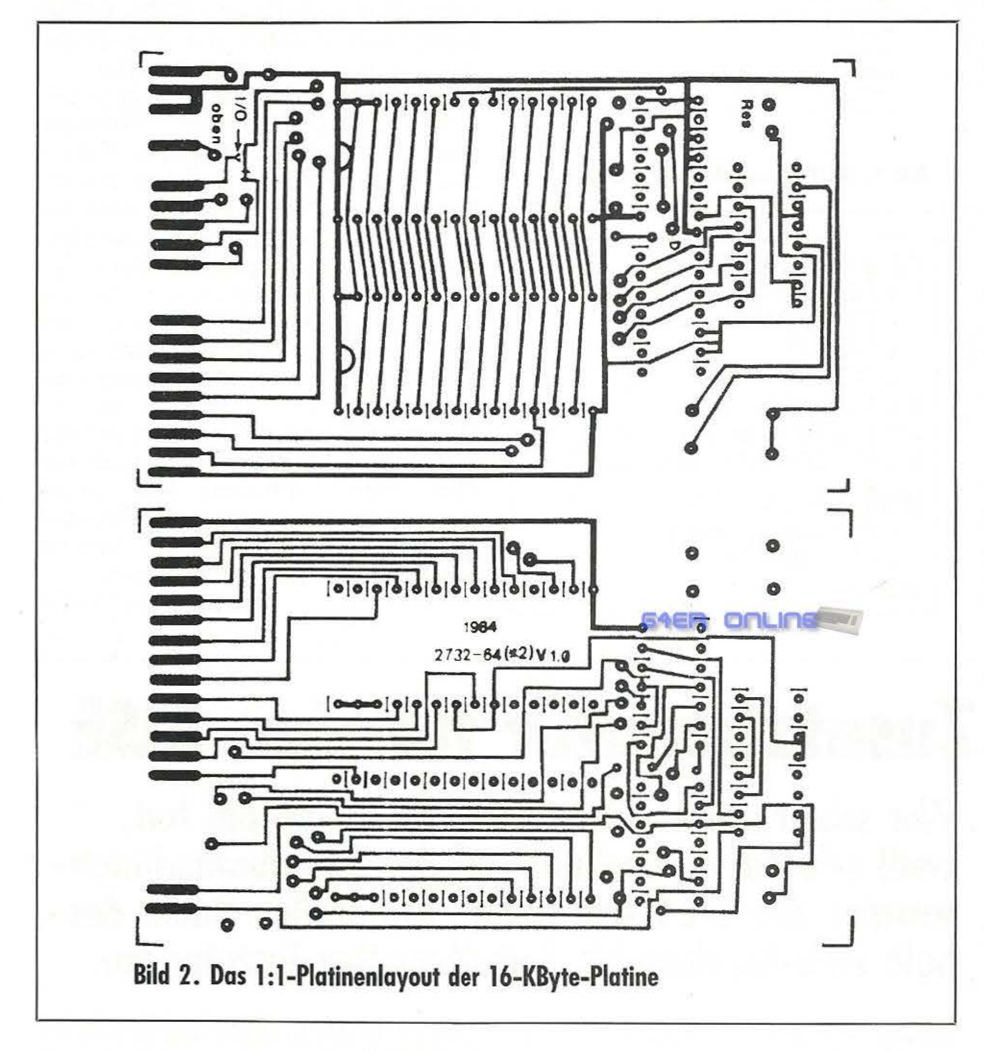


Bild 2. Das 1:1-Platinenlayout der 16-KByte-Platine

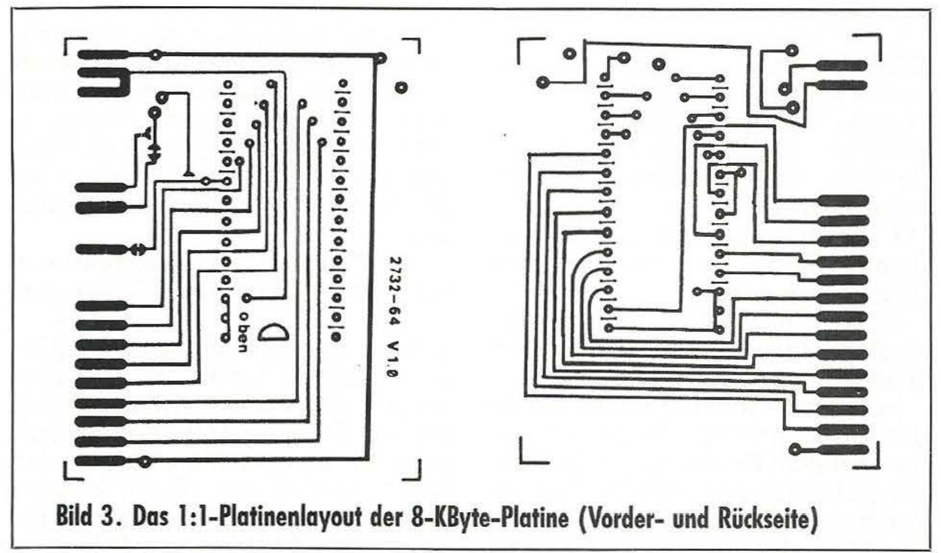


Bild 3. Das 1:1-Platinenlayout der 8-KByte-Platine (Vorder- und Rückseite)

hen, besser gesagt, daß sich immer nur eines in Betrieb befindet und zwischen beiden EPROMs umgeschaltet werden kann. Dazu benötigt man eine kleine Logikschaltung mit einem JK-Flip-Flop und zwei OR-Gattern (Bild 1). Beim JK-Flip-Flop werden die Eingänge J und K auf +5 V gelegt und bei jedem Wechsel von »High« nach »Low« an $\overline{T}$ (Takt) wir das Flip-Flop umgeschaltet. Der Ausgang Q wechselt dann von »0« nach »1«, »$\overline{Q}$« wechselt von »1« nach »0«. Man braucht noch eine Leitung, die vom Computer aus kontrolliert, beziehungsweise geschaltet werden kann und die mit dem Takt-Eingang des Flip-Flop verbunden wird. Zu diesem Zweck gibt es zwei I/O-Leitungen im C 64, nämlich $\overline{I/01}$ und $\overline{I/02}$, denen zwei Register im Computer entsprechen. Diese Register ($\overline{I/01}$: $DE00; $\overline{I/02}$: $DF00) kann man zum Beispiel mit POKE 57088,X oder STA $DE00 ansprechen. Dabei wechselt ihr Zustand kurzzeitig von »high« nach »low« und es entsteht gerade das Schaltverhalten, das man für das Flip-Flop benötigt. Die Zusammenhänge beim Schalten gehen am besten aus einer Logiktabelle hervor. Es darf dabei immer nur ein EPROM an den Pins $\overline{CE/OE}$ ein Signal »low« bekommen.

Einschalten (Hardware-Reset) beziehungsweise Zustand 1:
Q=L, $\overline{Q}$=H

A1	B1	Q1	A2	B2	Q2
Q=L	H=$\overline{ROML}$	H	$\overline{Q}$=H	H=$\overline{ROML}$	H
Q=L	L=$\overline{ROML}$	L	$\overline{Q}$=H	L=$\overline{ROML}$	H

Q1=L: EPROM A angewählt
Q2=H: EPROM B nicht angewählt

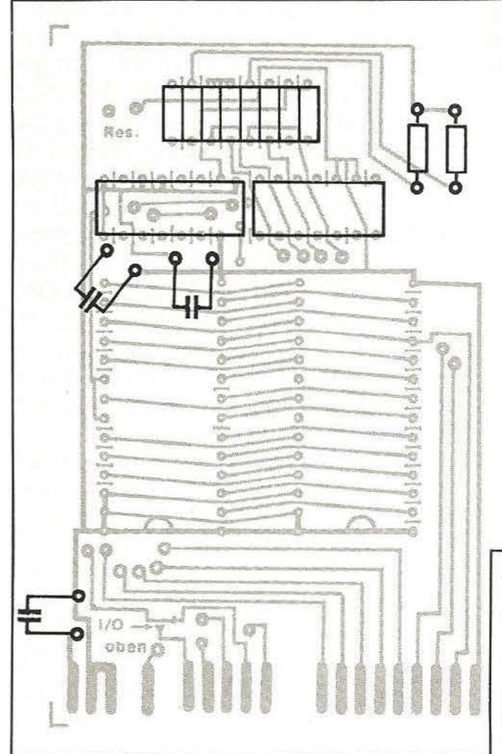
Sobald der Takt-Eingang für kurze Zeit von »High« auf »Low« geht, erhöht man den anderen Zustand des Flip-Flops.

Umschalten beziehungsweise Zustand 2: Q=H, $\overline{Q}$=L

A1	B1	Q1	A2	B2	Q2
Q=H	H=$\overline{ROML}$	H	$\overline{Q}$=L	H=$\overline{ROML}$	H
Q=H	L=$\overline{ROML}$	H	$\overline{Q}$=L	L=$\overline{ROML}$	L

Q1=H: EPROM A nicht angewählt
Q2=L: EPROM B angewählt

Die letzte Verwendungsmöglichkeit der Platine erlaubt das Ein- und Ausschalten von wahlweise einem 8-KByte-, oder 16-KByte-Bereich. Mit dem Schalter X5 wird zunächst $\overline{ROML}$ mit dem rechten EPROM verbunden ($8000). Für ein 16-KByte-Modul wird zusätzlich der Schalter S1 geschlossen und damit das linke EPROM im Bereich $A000 selektiert. Doch wie bereits beschrieben, greift der Computer immer noch nicht auf die EPROMs zu. Darum müssen die Leitung $\overline{EXROM}$ (zusätz-



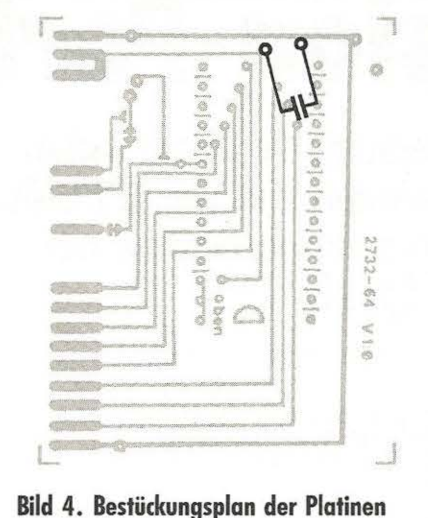


Bild 4. Bestückungsplan der Platinen

lich $\overline{\text{GAME}}$ bei 16 KByte) mit Masse verbunden werden. Setzt man in diese Verbindung nun das schon verwendete Flip-Flop, so können die Leitungen, je nach Zustand, von $\overline{\text{I/02}}$ auf Masse gelegt werden oder nicht. Das Ein- und Ausschalten erfolgt jetzt mit dem schon erwähnten Ansprechen von $\overline{\text{I/02}}$ ($DF00). Zu erwähnen ist noch, daß das Flip-Flop nach einem Hardware-Reset in den Grundzustand (Q=L, $\overline{\text{Q}}$=H) geht, da sein Reset-Eingang $\overline{\text{R}}$ mit dem Anschluß $\overline{\text{RESET}}$ des Expansion-Port verbunden ist. Dann ist im zweiten Fall das rechte EPROM selektiert und im letzten Fall das Modul eingeschaltet. Bei einem Software-Reset (SYS 64738) bleibt der Zustand des Flip-Flops, und damit der Platine, erhalten.

Bei der Eigenherstellung ist zu beachten, daß sich auf den Platinen einige Durchkontaktierungen (Bild 2 und 3) befinden. Außerdem sollten Sie darauf achten, ob die Bauteile mit der Platinenoberseite oder Unterseite verlötet werden müssen. Bestücken Sie die Platinen zuerst mit den Kondensatoren, und danach mit den Fassungen und Bauteilen (Bild 4). Achten Sie bei den Kondensatoren auf die richtige Polung. Bei der Verwendung von EPROMs des Typs 2732, die vier Pins weniger als ein 2764 haben, werden diese nicht bündig mit der Kerbe in die Fassung gesetzt. Die zwei Pins auf beiden Seiten der Kerbe bleiben frei.

(Markus Braun/aw)

Stückliste 2 x 8-KByte-Platine
2 x Fassung 28-polig
1 x 8-fach DIL-Schalter
1 x 74LS113
1 x 74LS32
2 x 4,7 µF Elko 16 V
1 x 10 nF Keramikkondensator
2 x 10 kOhm Widerstand ¼ V/A

Stückliste 1 x 8-KByte-Platine
1 x Fassung 28-polig
1 x 4,7 µ Elko 16 V

Stückliste

Der Hexer — Zusatztastatur für den MSE

Wer schon seitenweise Hex-Code eingetippt hat, weiß wie nervtötend das auf der Schreibmaschinentastatur des C 64 sein kann. Wir stellen Ihnen deshalb eine komfortable Selbstbau-Hex-Tastatur vor.

Als Programmierfan stehen Sie auch häufig vor dem Problem, seitenweise Zahlen eingeben zu müssen, zum Beispiel MSE-Listings. Meist erfolgen die Angaben im Hexadezimalsystem, das die Ziffern 0 bis F umfaßt. Deshalb zeigen wir Ihnen, wie man sich einfach eine Hex-Tastatur selbst bauen kann.

Der Bau dieser Hex-Tastatur dürfte dem geübten Bastler keine Schwierigkeiten bereiten. Dem handwerklich weniger Erfahrenen, können wir in Kürze eine Bezugsadresse für die entsprechende, fertig aufgebaute Tastatur geben.

Das wichtigste Kriterium der Tastatur war die Kompatibilität zu allen existierenden Programmen. Es mußte also eine Hardwarelösung geschaffen werden, die keinen Speicherplatz benötigt.

Der Verbindungsstecker zwischen Tastatur und C 64-Platine besitzt 19 Kontakte und eine sogenannte Codierungslücke. Diese Lücke in der Stiftleiste verhindert ein falsches Aufstecken. Die 19 Kontakte lassen sich in drei Gruppen aufteilen (Tabelle 1).

1. Die Kontakte 1 und 3 sind der RESTORE-Taste zugeordnet. Der zweite Kontaktstift fehlt. Die entsprechende Bohrung im Stecker ist verschlossen. Diese Codierung verhindert ein falsches Aufstecken des Tastatursteckers.

2. Die Stifte 5 bis 12 sind an die acht Reihen (Rows) der Tastaturmatrix (Bild 1) angeschlossen.

3. Die Stifte 13 bis 20 geben den logischen Zustand der acht Matrixspalten (Columns) wieder.

Jede Taste verbindet eine Spalte der Matrix mit einer Reihe der Matrix. Bei acht Reihen und acht Spalten gibt es folglich 8x8=64 Möglichkeiten der Verschaltung, die der C 64 vollständig ausnutzt. Bild 1 zeigt, welche Matrixpunkte die einzelnen Tasten einnehmen.

Alle Tastenkontakte sind im Ruhezustand geöffnet und werden durch Tastendruck geschlossen. Will man eine Taste von außen ersetzen, braucht nur die entsprechende Matrixspalte und Reihe über einen Taster miteinander verbunden zu werden.

Der Anschluß der Zusatztastatur erfolgt über eine Adapterleiste (Bild 2).

Nach dem Öffnen des C 64 (Garantieverlust!) wird der Tastaturstecker abgezogen und zusammen mit dem Zwischenstück wieder aufgesteckt. Im einzelnen besteht der Adaptersockel aus einem kleinen Stück Lochrasterplatine mit Kupferbahnen im Abstand von 2,54 mm.

Diese Platine stellt eine mechanisch feste Verbindung zwischen der Buchsenleiste (2,54 mm Rastermaß) und der Stiftleiste her. Die Buchsenleiste wird dann auf die Stiftkontakte der C 64-Platine gesteckt und der Tastaturstecker auf die Stiftkontakte des Adaptersteckers.

Es muß dann nur noch Stift 2 des Adaptersockels mit einem Seitenschneider abgezwickt und das Flachbandkabel angelötet werden. Das geschieht zwischen den Lötaugen der Buchsenleiste und den Lötstellen der Stifte.

Die Tastatur selbst besteht im wesentlichen aus einer Lochrasterplatine mit 24 Tasten. Bild 3 zeigt unseren Prototyp, bei dem noch einige Tasten »von Hand« beschriftet sind. Es hat sich jedoch eingebürgert, die 16 Hex-Tasten im Quadrat anzuordnen. Die Verschaltung der Tasten erfolgt mit dünnem isolierten Draht. Welche Pins der Stiftleiste des C 64 mit welchen Tasten verbunden werden müssen, entnehmen Sie bitte Bild 1.

Die Verpackung

Der Tastenblock mit der Lochrasterplatine kann in einem Gehäuse eingebaut werden. Man sollte ein Pultgehäuse wählen, so daß das Tastenfeld nach hinten hin ansteigt.

Bevor Sie sich nun auf die Suche nach Tasten machen, ein Tip: Billig kommt man an Tasten, indem man alte Computer-Tastaturen ausschlachtet. Achten Sie dabei aber darauf, daß die Tasten einzeln verwendbar sind und von der Grundplatte gelöst werden können. Die Kosten für neue Tasten mit Lochrasterplatte und Gehäuse dürften etwa 50 Mark betragen. Eine Stückliste finden Sie in Tabelle 2. Häufig findet man sogar schon fertig aufgebaute Tastenblöcke, die man nur noch richtig verschalten muß.

(Stefan Thienel/hm)

	13	14	15	16	17	18	19	20
12	DEL	£	+	9	7	5	3	1
11	RET.	*	P	I	Y	R	W	–
10	⇄	;	L	J	G	D	A	CTRL
9	↓↑	/	,	N	V	X	Shift links	STOP
8	F1	Shift rechts	.	M	B	C	Z	SPACE
7	F3	=	:	K	H	F	S	C=
6	F5	↑	@	O	U	T	E	Q
5	F7	HOME	—	0	8	6	4	2

Bild 1. Die Schaltmatrix der C 64-Tastatur. Zum Anschluß entnehmen Sie bitte die beiden Kontakte einer Taste der Schaltmatrix und verbinden Sie über den entsprechenden Taster



Bild 2. Querschnitt (längs einer Leiterbahn) des Adaptersteckers

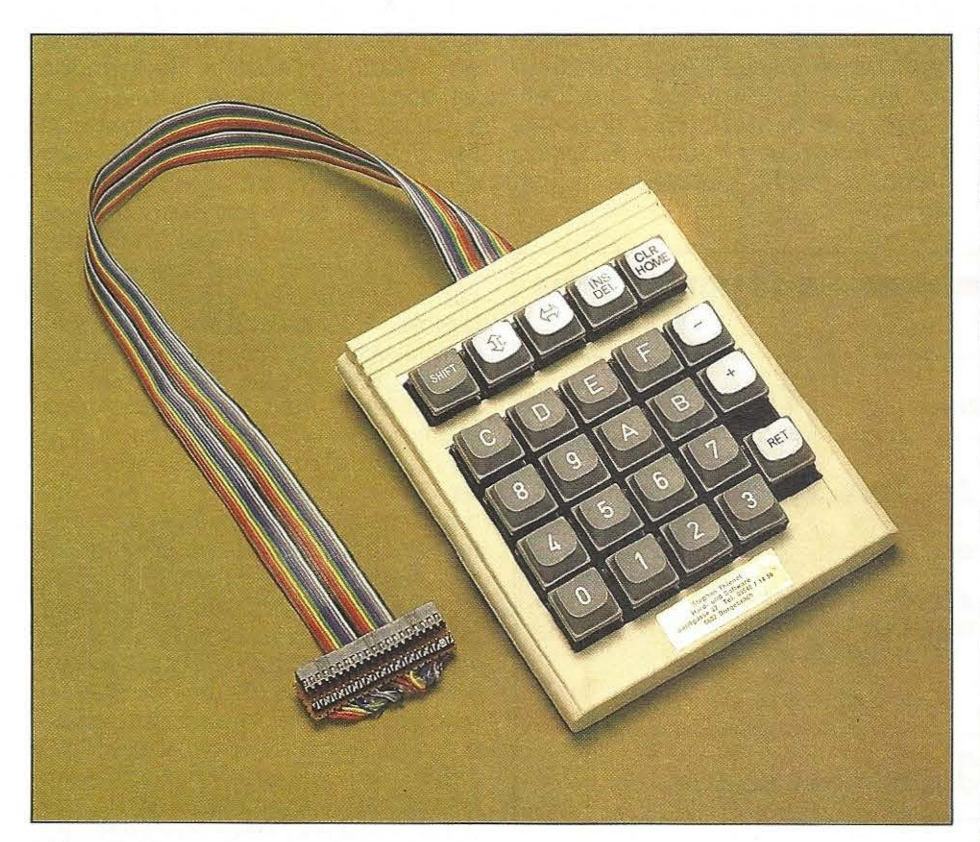


Bild 3. Ein Prototyp der Hex-Tastaturen. Eine von uns testweise zusammengebaute Version. Für die RETURN-Taste sollte eine doppeltgroße Tastenkappe die Eingabe erleichtern.

24 Computer-Tasten 1 x EIN
24 Tastenkappen mit/ohne Beschriftung
1 Lochrasterplatine 11x13 cm
1 Lochrasterplatine mit Kupferstreifen; 20 Bahnen à 2 cm Länge
3 m Schaltdraht, verschiedene Farben
1 m Flachbandkabel 20adrig
1 Stiftleiste, 20polig
1 Buchsenleiste 20polig
1 Gehäuse, nach hinten ansteigend

Tabelle 2. Die Stückliste zur Hex-Tastatur

20 ○ … 13 ○ — 8 Spalten (COL0...COL7)
12 ○ … 5 ○ — 8 Reihen (ROW0...ROW7)
+5V → ○
○ ← Restore
Code → x
○ ← Restore

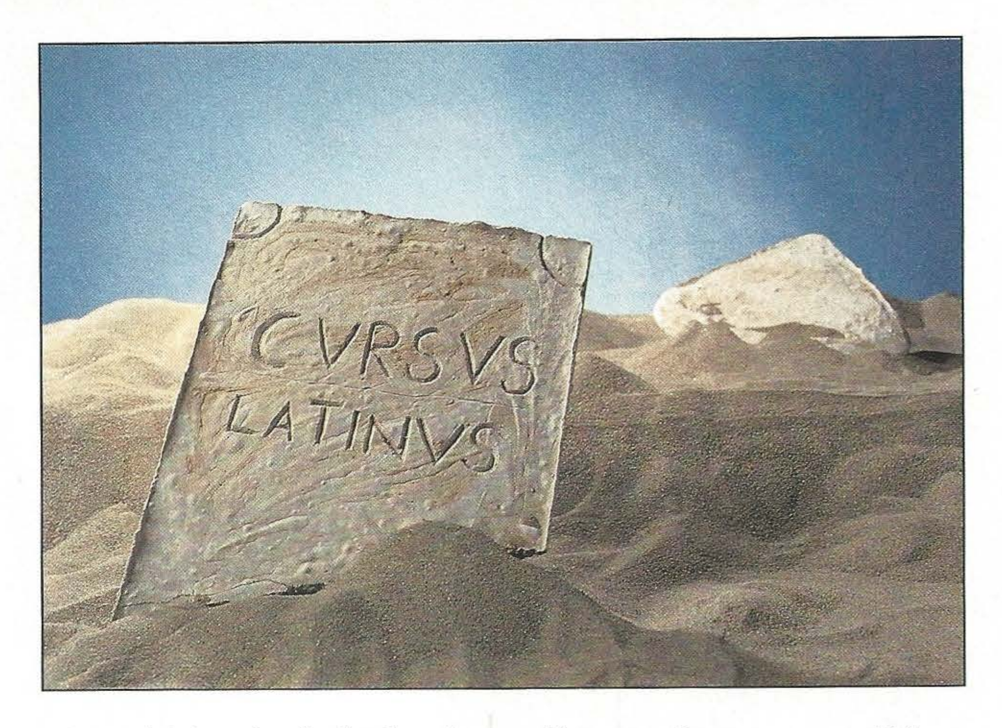
Tabelle 1. Die Pinbelegung der Tastatur-Stiftleiste auf der C 64-Platine





Fit in Latein mit dem C 64

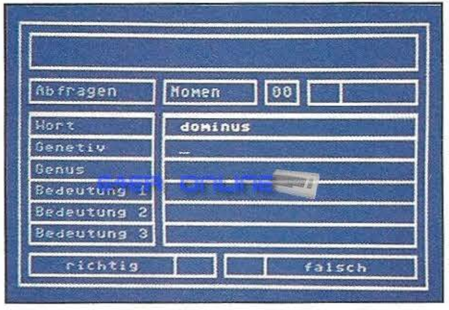
Ein leidiges Thema für alle Lateinschüler: Vokabeln pauken. Mit dem C 64 kann es fast schon Spaß machen, vor allem, wenn man ein speziell auf lateinische Vokabeln zugeschnittenes Vokabellernprogramm hat.



Lernprogramme gibt es ja mittlerweile recht viele, aber keines, das zum Beispiel die verschiedenen lateinischen Fälle berücksichtigt. »Cursus-Latinus« hat neben dieser Besonderheit noch eine beispielhafte Benutzerführung.

Die Vokabeln lassen sich in verschiedene Lektionen aufteilen. Diese kann man später beliebig erweitern oder verbessern.

Bei der Eingabe der einzelnen Vokabeln muß man zuerst die Wortart wählen. Entweder »Nomen«, oder »Adjektiv«, »Adverb«, »Verb«, »Deponens« oder »Semideponens«. Hier zeigen sich bereits die Stärken des Programms. Bei »Nomen« kann man dann das lateinische Wort selber,



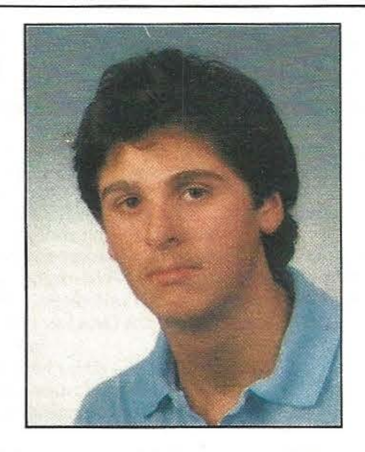
»Cursus Latinus« beim Abfragen

dessen Genitiv, den Genus und maximal drei deutsche Bedeutungen eingeben.

Hat man »Deponens« gewählt, so besteht die Eingabe aus der ersten und der zweiten Stammform, und wieder den drei deutschen Bedeutungen. Beim Adjektiv muß zusätzlich noch der Genus eingegeben werden. Später beim Abfragen erteilt das Programm auf jede richtige Eingabe einen Punkt. Diese werden nach jeder Lektion ausgewertet und ausgegeben.

Im Programm wurde auf jede unnötige grafische Spielerei verzichtet. Nicht gespart wurde dagegen an aufwendigen Bildschirmmasken und Hilfstafeln, die auf einen Tastendruck eine Übersicht der einzelnen Tastenfunktionen bringen.

(Bernd Kraus/tr)



Ich wurde am 23. Dezember 1969 in Heidelberg geboren und besuche zur Zeit die zehnte Klasse des Gymnasiums Walldorf. Meine Vorliebe für Informatik fand ich Ende der sechsten Klasse auf einem Schulfest unserer Schule.

Damals zeigten einige Teilnehmer der Arbeitsgemeinschaft Computer ihr Können auf einigen alten Commodore-Computern. Als ich den zuständigen Lehrer bat, mich auch in einer solchen Arbeitsgemeinschaft teilnehmen zu lassen, vertröstete er mich auf Anfang der nächsten Klasse und in dieser dann auf die folgende. Als mir das Spiel zu dumm wurde, beschloß ich, mir selbst einen guten, gängigen Computer anzuschaffen. Einen VC 20. Ich programmierte eifrig. Die Ergebnisse waren mehr als kläglich und die Kassetten, auf denen ich meine Programme speicherte, wurden schnell wieder mit Musik bespielt. Als ich im Herbst 1983 erstmals vom C 64 hörte, war ich nicht sehr überzeugt, daß dieser Computer dem VC 20 ernsthaft Konkurrenz machen könne. Dieser Glaube wurde auch dadurch bestätigt, daß ich in einem Kaufhaus ein Programm namens »Geisterjäger« eintippte und sah, daß man dieses Programm auch leicht auf einem VC 20 schreiben konnte. Erst als sich Anfang 1984 ein Freund einen C 64 kaufte und mir vorführte, wie man einen buntes Ballon-Sprite von der einen Bildschirmseite auf die andere und wieder zurück POKEn konnte, was für schöne Klänge diese Maschine vollbrachte und wie toll es doch ist, mit prächtigen 40 Zeichen pro Zeile statt 22 zu arbeiten, entschloß ich mich, meinen alten VC 20 loszuwerden und mir schnell das »Wunderding« C 64 anzuschaffen, wenn möglich, mit Floppy-Laufwerk. Das tat ich dann auch. Ich schrieb viele Programme, die natürlich erheblich besser waren als die, die ich auf dem VC 20 geschrieben hatte.

In dieser Zeit entstand auch das Programm »Cursus Latinus«, das ich in Verärgerung darüber geschrieben hatte, daß es kein Vokabellernprogramm gab, das auch für lateinische Vokabeln von Nutzen war. Mein Programm sollte mit möglichst wenigen Farben und einer komfortablen Eingabe-Maske arbeiten.

Bei der Entwicklung stieß ich auf einige Schwierigkeiten, wie dem Problem, daß beim zufälligen Abfragen der Vokabeln kein Wort wiederholt wird.

Motiviert von einem Klassenkameraden, der das leidige Pech hatte, Französisch als zweite Fremdsprache zu wählen, arbeite ich nun an einer Version mit französischen Vokabeln.

(Bernd Kraus)



Checksummer 64 V3 und MSE

Bei beiden Programmen handelt es sich um Eingabeerleichterungen unserer Listings. Ferner werden alle Tippfehler bemerkt und angezeigt.

Mit dem Checksummer 64 V3 läßt sich überprüfen, ob eine Zeile korrekt eingegeben wurde. Dazu muß zuerst das Programm Checksummer 64 V3 geladen und mit RUN gestartet werden. Ist das geschehen, erscheint, sobald Sie eine Zeile eingeben und mit RETURN abschlossen haben, links oben auf dem Bildschirm eine geklammerte Zahl in reverser Darstellung. Bei dieser Zahl handelt es sich um eine Prüfsumme. Sie muß mit der ebenfalls geklammerten Zahl am Ende jeder Basic-Zeile übereinstimmen. Tut sie das nicht, haben Sie einen Tippfehler gemacht. Die Prüfsummen im Listing dürfen nicht mit abgetippt werden. Basic-Listings enthalten keinerlei Steuer- und Grafikzeichen, dafür aber unter- beziehungsweise überstrichene Zeichen und Wörter in geschweiften Klammern. Dabei bedeuten:
unterstrichenes Zeichen = SHIFT-Taste + Zeichen
überstrichenes Zeichen = Commodore-Taste + Zeichen
die Wörter in geschweiften Klammern:
UP, DOWN, RIGHT, LEFT = Cursorsteuertasten

Ferner werden alle Farben ausgeschrieben. Die Farbe, deren Abkürzung Sie auf der Tastatur des C 64 wiederfinden, erreichen Sie über die Control-Taste (CTRL) beziehungsweise Commodore-Taste plus der Farbtaste

Zahlen in geschweiften Klammern geben an, wie oft die Tastenkombination unmittelbar hinter der Zahl zu drücken ist, zum Beispiel 2SPACE = 2x Leertaste.

Das Listing zu diesem Checksummer V3 finden Sie zum erstenmal in der Ausgabe 8/85, Seite 54, und dem Drucker/ Grafik-Sonderheft. Alle anderen Checksummer-Listings (V2) sind damit hinfällig. Die ausführliche Anleitung aus diesen Ausgaben kann jedoch weiterhin benutzt werden.

Der **MSE** unterstützt die Eingabe von Listings, die in Maschinensprache geschrieben wurden. Bei ihm handelt es sich um ein Maschinensprache-Editor, der Fehleingaben ausschließt. Eine abgetippte Zeile wird nur dann angenommen, wenn sie richtig eingegeben wurde.

Das Listing zum MSE mit ausführlicher Anleitung finden Sie in den Ausgaben 1/85 bis 6/85. Auch dieses Programm ist auf jeder Leserservice-Diskette gespeichert.

Außerdem schicken wir Ihnen gegen Einsendung eines an Sie selbst adressierten Briefumschlags (Größe DIN C5 = 0,80 Mark, DIN C4 = 1,10 Mark Porto) gerne das Listing zum MSE und Checksummer 64 V3 zu.

C 64-Schreiberling — Drucken wie gemalt

Mit einem grafikfähigen Drucker sind der Kreativität kaum Grenzen gesetzt. Phantasievolle Schriften, wie sie jeder aus Märchen kennt, sind eines der reizvollen Anwendungsgebiete dieser Drucker. Mit dem Programm »Schreiberling« ersetzen sie sogar Tusche und Federhalter.

Wer hat nicht schon die wundervoll geschwungenen Schriften alter Märchenbücher bewundert und sich gewünscht, doch so schreiben zu können? Die Erfahrung zeigt aber, daß es gar nicht so einfach ist, mit Federhalter und Tusche den richtigen Schwung herauszubekommen, ohne dabei in Tintenklecksen unterzugehen. Wer aber trotzdem schöne Grußkarten, Einladungen oder sogar Urkunden in einer phantasievollen Schrift erstellen möchte, ist auf die Hilfe des Computers angewiesen. Zusammen mit dem Programm »Schreiberling« wird es Ihnen sicherlich oft gelingen, Ihre Verwandten und Freunde mit gestochen scharfen Schriftstücken zu überraschen.

»C 64-Schreiberling« verwendet eine der interessantesten Fähigkeiten eines Druckers — seinen Grafikdruck. Jeder, der einen Drucker mit dieser Funktion besitzt, hat sicherlich schon im Handbuch über die Technik des Binärausdrucks gelesen. Was man aber, außer Hardcopies, damit noch alles machen kann, soll hier beschrieben werden. Das Programm wurde für einen CP-80X entwickelt. Da dieser aber die gleichen Steuerzeichen wie ein Epson RX/FX-80 verwendet, funktioniert »C 64-Schreiberling« auch mit einer ganzen Reihe anderer Drucker.

Das Programm »Schreiberling« (siehe Listing) verwendet den Binärzeichendruck. Dabei hängt das Aussehen eines Buchstabens nicht mehr vom vorprogrammierten Inhalt des Festwertspeichers, sondern von einer eigenen Definition ab. Dazu ist es notwendig, dem Drucker genau zu befehlen, welche Nadel wann und wo drucken soll, um das gewünschte Resultat zu erhalten. Beim Binärzeichendruck werden nur acht der neun vorhandenen Nadeln des Druckers angesprochen. Die Nadeln sind übereinander angeordnet. Die unterste Nadel hat den Wert 1, die oberste den Wert 128 (Bild 1). Um nun ein Zeichen zu erhalten, muß man dem Drucker für jede Nadelstellung einzeln befehlen, welche Nadeln aktiv sein sollen.
Der Druckkopf rückt danach von selbst um ein kleines Stück weiter und druckt die nächste Nadelkombination. Auf diese Art werden beispielsweise Hardcopies vom Bildschirm gedruckt.

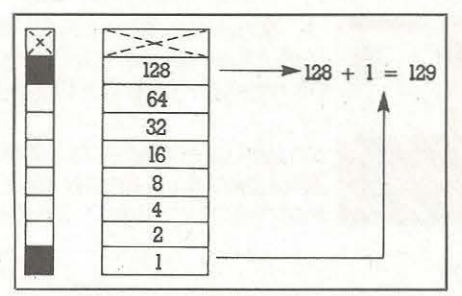


Bild 1. Jede der einzelnen Nadeln eines Druckkopfes hat ihren eigenen Zahlen-Wert

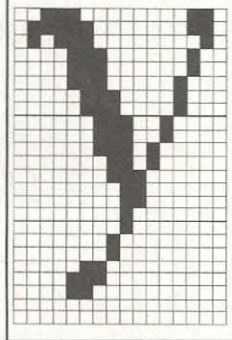


Bild 2. Jeder Buchstabe besteht aus einer 16x24-Punktematrix

Oder man entwirft einen Satz geänderter Zeichen, wie es bei dem Programm Schreiberling gemacht wurde.

Betrachten wir einmal das große »Y« mit Schreiberling ausgedruckt (Bild 2). Man sieht, daß es eine 16 x 24-Punktematrix besitzt. Da der Drucker aber nur acht Nadeln verwendet, muß er insgesamt dreimal drucken, um das »Y« zusammenzusetzen. Zuerst die Oberteile aller Zeichen, dann die Mittelteile und zuletzt die Unterteile.

Nach dem Eingeben und Starten des Programms dauert es eine Weile, bis alle DATA-Zeilen gelesen sind. Danach erscheint auf dem Bildschirm: ».....eine.Zeile.eingeben:.......«. Das heißt, Sie können nun einen beliebigen Text, mit einer Länge bis zu einer Druckzeile, eingeben. Da die einzelnen Zeichen unterschiedlich lang sind (das I ist beispielsweise kürzer als das M), läßt sich nicht genau sagen, wieviele Zeichen in eine Druckzeile passen. Auf jeden Fall sollten Sie niemals zu viele Zeichen eingeben, denn sie werden nicht mehr gedruckt. Bei normaler Dichte (normales Drucken mit Schreiberling) passen ungefähr 40 Zeichen in eine Druckzeile. Am besten ist es, Sie probieren es einfach einmal aus. Geben Sie dazu einen Bei-

The never ending Story

Bild 3. So sollte Ihre erste Schriftprobe aussehen

spieltext, etwa: »The never ending Story« ein. Dieser Text wird nun in normaler Dichte ausgedruckt (Bild 3). Das ist aber noch nicht alles, was Schreiberling kann. Wenn Sie vor die Zeile ein »#« stellen, werden alle Zeichen mit doppelter Dichte ausgegeben, probieren Sie es aus. Das Ergebnis sollte ähnlich wie Bild 4 auf Ihrem Drucker erscheinen.

DAS MAERCHEN VOM KOENIG MURDAS
von Stanislaw Lem

Nach dem guten Koenig Helixander bestieg sein Sohn Murdas den Thron. Alle haermten sich darob, denn jener war ehrsuechtig und schreckhaft. Er hatte beschlossen, sich den Beinamen DER GROSSE zu verdienen, und fuerchtete sich dabei vor Zugluft,Geistern, Wachs, da man auf gewachstem Parkett ein Bein brechen kann, Verwandten, denn die stoeren beim Regieren, am meisten aber vor Weissagungen. ...

Bild 4. Nicht nur Märchen lassen sich drucken

Für alle, die gerne etwas experimentieren wollen, hier noch ein paar Hinweise. Beim Abtippen können Sie alle REM-Zeilen weglassen. Das Programm beginnt mit Zeile 145, in der die Bildschirmfarben verändert werden. In Zeile 160 werden drei Felder dimensioniert, in die später die Zeichendaten (Ober-, Mittel- und Unterteil) eingelesen werden. Zeile 200 bis 290 füllt die drei Felder. In Zeile 300 bis 360 befindet sich der Einzeleditor und in den Zeilen 422 bis 550 werden die Daten auf den Drucker ausgegeben. Die I-Schleife von 422 bis 550 liest die Buchstaben vom Bildschirm (bei normaler Dichte eine Zeile, bei doppelter Dichte ungefähr zwei Zeilen). Die J-Schleife läßt die I-Schleife dreimal die Buchstaben lesen, da die Ober-, Mittel- und Unterteile einzeln gedruckt werden. In den Zeilen 1 000 bis 2 460 stehen die Zeichendaten. Da das Programm ganz in Basic geschrieben wurde, läßt es sich zwar leicht editieren, dafür ist es nicht gerade schnell — aber wozu gibt es Compiler? (K. Kramer/aw)

```
5 REM ------ SCHREIBERLING -------------------         <156>
10 :                                                  <242>
20 REM ----- NACHHILFEUNTERRICHT -------------        <150>
21 :                                                  <253>
22 REM ----- ANGELSAECHSISCH FUER DRUCKER ----        <160>
23 :                                                  <255>
100 REM ---- ALLE REM-ZEILEN KOENNEN AUSGELASSEN WER
    DEN                                               <166>
110 :                                                 <086>
120 REM VON KLAUS KRAMER                              <067>
130 REM     OLAFSTR. 20 A                             <204>
140 REM     4950 MINDEN                               <090>
145 POKE 53281,14:POKE 53280,14:POKE 646,6            <070>
150 PRINT CHR$(147);"SCHREIBERLING SUCHT DATEN-BITTE
     WARTEN."                                         <225>
160 DIM A1$(72),A2$(72),A3$(72)                       <168>
162 FOR I=1 TO 16:Q$=Q$+CHR$(0):NEXT I                <044>
163 FOR I=1 TO 23:O$=O$+CHR$(0):NEXT I                <098>
170 :                                                 <146>
200 REM EINLESEN-----------------------               <076>
210 :                                                 <186>
220 RESTORE                                           <014>
230 FOR I=1 TO 72                                     <125>
240 FOR J=1 TO 3                                      <091>
245 :  READ X                                         <158>
250 :  IF X=-1 THEN 280                               <007>
260 :  IF J=1 THEN A1$(I)=A1$(I)+CHR$(X)              <213>
265 :  IF J=2 THEN A2$(I)=A2$(I)+CHR$(X)              <029>
267 :  IF J=3 THEN A3$(I)=A3$(I)+CHR$(X)              <098>
270 :  GOTO 245                                       <058>
280 NEXT J                                            <118>
282 IF LEN(A3$(I))=0 THEN A3$(I)=LEFT$(O$,LEN(A1$(I)
    ))                                                <138>
285 NEXT I                                            <115>
290 :                                                 <012>
300 REM ZEILEN-TEXTEDITOR---------------              <100>
310 :                                                 <032>
320 PRINT CHR$(147);CHR$(14);CHR$(18);".....EINE.ZEI
    LE.EINGEBEN:.............";                       <115>
325 PRINT CHR$(146)                                   <228>
330 GET T$:PRINT T$;                                  <039>
340 IF T$=CHR$(13)THEN 410                            <176>
350 GOTO 330                                          <088>
360 :                                                 <082>
400 REM AUSGABE-------------------------              <126>
405 :                                                 <127>
410 OPEN 1,4:E$=CHR$(27)                              <087>
411 Y=50:P=0                                          <184>
414 Z=PEEK(1064)                                      <117>
415 IF Z=35 THEN Y=99:P=1                             <174>
417 PRINT#1,E$"A"CHR$(8);:PRINT#1,E$"U"CHR$(1);       <033>
418 FOR J=1 TO 3                                      <015>
420 :  IF Z=35 THEN PRINT#1,E$"L"CHR$(0)CHR$(5);      <109>
421 :  IF Z<>35 THEN PRINT#1,E$"K"CHR$(128)CHR$(2);   <195>
422 :  FOR I=1064+P TO 1064+Y                         <057>
425 :     W=PEEK(I)                                   <167>
430 :     IF W=32 OR W=96 THEN PRINT#1,Q$;            <225>
440 :     IF W=63 THEN F=-17:GOSUB 910                <241>
450 :     IF W=40 OR W=41 THEN F=4:GOSUB 910          <114>
460 :     IF W=33 THEN F=10:GOSUB 910                 <162>
470 :     IF W>64 AND W<91 THEN F=-64:GOSUB 910       <110>
475 :     IF W>0 AND W<27 THEN F=46:GOSUB 910         <184>
480 :     IF W>43 AND W<60 THEN F=-17:GOSUB 910       <081>
500 :                                                 <222>
550 :  NEXT I:IF J<3 THEN PRINT#1                     <150>
560 NEXT J                                            <144>
670 PRINT#1:CLOSE 1:GOTO 320                          <229>
890 :                                                 <104>
900 REM UNTERPROGRAMM 1----------------               <007>
905 :                                                 <119>
910 IF J=1 THEN PRINT#1,A1$(W+F);                     <166>
920 IF J=2 THEN PRINT#1,A2$(W+F);                     <216>
930 IF J=3 THEN PRINT#1,A3$(W+F);                     <010>
940 RETURN                                            <236>
1000 REM GROSS A-Z ---------------------              <024>
1002 :                                                <216>
1010 DATA 0,32,65,198,200,240,252,127,31,1,0,0,0,0,0
     ,0,-1                                            <223>
1020 DATA 0,62,255,3,3,2,4,200,248,254,31,3,1,2,4,0,
     -1,-1                                            <250>
1030 DATA 0,128,255,255,255,131,131,131,131,131,131,
     68,124,56,0,-1                                   <187>
1040 DATA 0,1,255,255,255,1,1,1,1,1,1,130,254,124,0,
     -1,-1                                            <216>
1050 DATA 0,7,31,63,112,192,128,128,128,128,128,128,
     64,120,56,0,-1                                   <035>
1060 DATA 0,224,248,252,14,3,1,1,1,1,1,1,2,30,28,0,-
     1,-1                                             <003>
1070 DATA 0,128,255,255,255,128,128,128,128,128,64,1
     12,63,31,15,0,-1                                 <251>
1080 DATA 0,1,255,255,255,1,1,1,1,1,2,14,252,248,240
     ,0,-1,-1                                         <086>
1090 DATA 0,7,31,63,114,194,130,130,131,131,131,131,
     66,112,48,0,-1                                   <035>
1100 DATA 0,224,248,252,14,3,1,1,1,1,129,1,2,14,12,0
     ,-1,-1                                           <212>
1110 DATA 0,128,255,255,255,66,66,130,130,131,131,13
     1,131,98,96,0,-1                                 <049>
1120 DATA 0,0,255,255,255,0,0,0,0,0,0,128,0,0,0,0,-1 <057>
1130 DATA 4,8,248,240,192,0,0,0,0,0,0,0,0,0,0,-1     <050>
1140 DATA 0,7,15,63,112,192,128,128,128,128,194,99,9
     9,3,2,0,-1                                       <247>
```

Listing. Schreiberling druckt phantasievolle Schriften. Bitte beachten Sie die Eingabehinweise auf Seite 54.

```
1150 DATA 0,248,252,254,7,3,1,1,2,4,8,255,255,255,0,
     0,-1                                             <077>
1160 DATA 0,0,0,0,0,0,0,0,4,8,248,240,192,0,0,-1      <100>
1170 DATA 0,128,255,255,255,2,4,8,8,16,16,24,31,15,3
     ,0,-1                                            <189>
1180 DATA 0,0,255,255,255,0,0,0,0,0,3,15,255,249,225
     ,0,-1,-1                                         <122>
1190 DATA 0,128,255,255,255,0,0,-1                    <011>
1200 DATA 0,0,255,255,255,1,0,-1,-1                   <055>
1210 DATA 0,0,128,255,255,255,0,-1                    <002>
1220 DATA 0,0,0,255,255,255,0,-1                      <233>
1230 DATA 0,4,8,248,240,192,0,-1                      <191>
1240 DATA 0,128,255,255,255,3,3,6,4,12,56,240,224,0,
     0,0,-1                                           <150>
1250 DATA 0,1,255,255,255,0,192,240,124,30,3,1,1,1,6
     ,0,-1,-1                                         <080>
1260 DATA 0,128,255,255,255,0,0,0,0,0,0,0,0,0,0,-1    <015>
1270 DATA 0,1,255,255,255,1,1,1,1,1,1,1,1,3,7,0,-1,-
     1                                                <144>
1280 DATA 0,7,31,56,96,192,128,128,128,128,127,63,12
     7,128,128,128,128                                <091>
1290 DATA 192,96,56,31,7,0,-1                         <112>
1300 DATA 0,192,248,62,7,1,1,0,0,0,255,255,255,0,0,0
     ,1,1,7,62,248,192,0,-1,-1                        <041>
1310 DATA 0,128,255,255,255,63,3,0,0,0,0,128,255,255
     ,0,0,-1                                          <216>
1320 DATA 0,0,255,255,255,0,224,240,56,12,6,2,255,25
     5,1,0,-1                                         <106>
1330 DATA 4,8,248,240,192,0,0,0,0,0,0,0,0,0,0,-1      <252>
1340 DATA 0,7,31,63,96,64,128,128,128,128,64,96,63,3
     1,7,0,-1                                         <163>
1350 DATA 0,224,248,252,6,2,1,1,1,1,2,6,252,248,224,
     0,-1,-1                                          <195>
1360 DATA 0,128,255,255,255,48,112,192,128,128,128,1
     92,96,63,31,0,-1                                 <101>
1370 DATA 0,0,255,255,255,12,14,3,1,1,1,3,6,252,248,
     0,-1                                             <052>
1380 DATA 0,0,252,252,252,4,0,0,0,0,0,0,0,0,0,-1      <239>
1390 DATA 0,15,63,127,192,128,128,128,128,128,64,96,
     63,63,63,64,0,-1                                 <232>
1400 DATA 0,240,252,254,3,1,1,1,1,1,14,12,255,255,25
     5,0,0,-1                                         <113>
1410 DATA 0,0,0,0,0,0,0,0,0,0,4,252,248,248,0,0,-1    <242>
1420 DATA 0,128,255,255,255,129,129,129,129,129,195,
     126,60,0,0,0,-1                                  <254>
1430 DATA 0,1,255,255,255,0,128,192,96,56,30,7,3,3,2
     ,0,-1,-1                                         <185>
1440 DATA 0,30,63,127,199,131,131,131,131,131,131,13
     1,195,113,48,0,-1                                <101>
1450 DATA 0,28,30,3,129,129,129,129,129,129,131,199,
     254,252,248,0,-1,-1                              <020>
1460 DATA 0,32,96,192,128,128,255,255,255,128,128,12
     8,128,128,0,-1                                   <086>
1470 DATA 0,0,0,0,0,0,255,255,255,1,0,0,0,0,0,-1,-1   <151>
1480 DATA 0,7,159,255,128,128,0,0,0,0,128,255,255,25
     5,0,0,-1                                         <052>
1490 DATA 0,248,252,254,1,1,1,1,2,4,8,255,255,255,1,
     0,-1,-1                                          <036>
1500 DATA 0,96,192,224,252,127,31,3,0,0,0,0,7,216,22
     4,0,-1                                           <160>
1510 DATA 0,0,0,0,0,0,224,248,255,15,48,192,0,0,0,0,
     -1,-1                                            <056>
1520 DATA 0,3,15,191,192,128,0,0,3,15,191,192,128,0,
     0,0,0,128,255,255                                <002>
1530 DATA 255,0,0,-1                                  <119>
1540 DATA 0,248,254,254,3,1,1,1,250,254,254,3,1,1,1,
     2,2,4,255,255,255,1,0,-1,-1                      <063>
1550 DATA 0,32,64,192,240,252,63,15,3,4,8,16,224,192
     ,0,0,-1                                          <042>
1560 DATA 0,7,11,16,32,64,128,192,240,252,62,15,3,1,
     6,0,-1,-1                                        <086>
1570 DATA 0,96,192,224,252,127,31,3,0,0,0,0,7,216,22
     4,0,-1                                           <232>
1580 DATA 0,0,0,0,0,0,224,248,255,14,48,192,0,0,0,0,
     -1                                               <203>
1590 DATA 0,0,0,0,12,28,24,96,128,0,0,0,0,0,0,0,-1    <062>
1600 DATA 0,128,192,192,224,128,128,129,131,143,158,
     188,240,224,192,0,-1                             <063>
1610 DATA 0,3,7,15,29,121,241,225,129,1,1,7,3,3,1,0,
     -1,-1                                            <097>
1611 :                                                <063>
1612 REM  ,-./  0 BIS 9 ------------------            <083>
1613 :                                                <065>
1614 DATA 0,0,0,0,0,-1,0,0,3,3,0,-1,0,48,192,0,0,-1   <251>
1615 DATA 0,0,1,1,1,1,1,1,1,1,1,1,0,0,-1             <082>
1616 DATA 0,0,128,128,128,128,128,128,128,128,128,12
     8,0,0,-1,-1                                      <062>
1617 DATA 0,0,0,0,0,0,-1,0,0,3,3,0,0,-1,-1            <114>
1618 DATA 0,0,0,0,0,0,3,12,48,192,0,0,-1              <013>
1619 DATA 0,0,3,12,48,192,0,0,0,0,0,0,-1,-1           <163>
1620 DATA 0,0,3,12,16,32,32,32,48,56,15,,0,-1         <168>
1630 DATA 0,240,252,14,3,1,1,1,2,12,240,0,0,-1,-1     <241>
1640 DATA 0,16,31,63,63,0,0,-1                        <210>
1650 DATA 0,0,255,255,254,2,0,-1,-1                   <003>
1660 DATA 0,8,16,32,32,32,48,31,15,0,0,0,0,0,-1       <013>
1670 DATA 0,1,3,7,15,29,249,225,193,1,1,3,2,0,-1      <218>
1680 DATA 0,0,0,128,128,128,128,128,128,128,0,0,0,0,
     -1                                               <125>
1690 DATA 0,16,32,33,51,62,28,0,0,-1                  <003>
1700 DATA 0,64,128,128,128,255,254,56,0,-1            <146>
1710 DATA 0,16,32,64,128,128,0,0,0,-1                 <169>
1720 DATA 0,0,0,0,1,2,4,15,31,63,0,0,0,0,-1           <141>
1730 DATA 0,32,96,160,32,32,32,255,255,254,32,32,32,
     0,-1,-1                                          <192>
1740 DATA 0,1,7,59,51,19,19,19,17,16,0,-1             <199>
1750 DATA 0,128,0,0,0,131,252,240,0,0,-1              <016>
1760 DATA 0,8,16,32,64,128,0,0,0,0,-1                 <017>
1770 DATA 0,3,15,18,36,68,76,140,12,7,3,0,-1          <107>
1780 DATA 0,248,254,7,1,1,1,1,6,248,0,-1,-1           <075>
1790 DATA 0,48,48,24,24,24,17,22,56,32,0,-1           <212>
1800 DATA 0,0,1,15,30,112,128,0,0,0,0,-1              <033>
1810 DATA 0,48,240,192,0,0,0,0,0,0,-1                 <025>
1820 DATA 0,0,121,135,131,131,133,121,0,0,-1          <045>
1830 DATA 0,124,194,129,1,1,129,226,124,0,-1,-1       <002>
1840 DATA 0,0,3,4,4,8,8,12,6,7,1,0,-1                 <050>
1850 DATA 0,248,252,6,3,1,1,2,2,229,254,0,-1          <100>
1860 DATA 0,0,0,0,0,8,16,32,64,128,0,0,-1             <154>
1862 :                                                <060>
1864 REM -- SATZZEICHEN :;!()?----------              <160>
1866 :                                                <064>
1870 DATA 0,0,3,3,0,0,-1,0,0,3,3,0,0,-1,-1            <248>
1880 DATA 0,0,0,0,0,0,-1,0,0,51,51,0,0,-1,0,48,192,0
     ,0,0,-1                                          <233>
1910 DATA 0,0,127,255,254,0,0,-1,0,0,243,243,0,0,0,-
     1,-1                                             <204>
1920 DATA 0,31,63,64,128,0,-1,0,248,252,2,1,0,-1,-1   <036>
1930 DATA 0,128,64,63,31,0,-1,0,1,2,252,248,0,-1,-1   <063>
1940 DATA 0,48,112,192,129,135,252,120,0,-1           <240>
1950 DATA 0,0,0,115,243,0,0,0,0,-1,-1                 <026>
1952 :                                                <150>
1954 REM KLEIN A-Z                                    <049>
1956 :                                                <154>
1960 DATA 0,0,1,3,2,2,2,1,0,1,3,2,0,0,-1              <085>
1970 DATA 0,252,254,3,1,1,1,1,130,254,255,1,2,0,-1,-
     1                                                <086>
1980 DATA 0,64,255,255,0,1,2,2,2,2,1,1,0,-1           <106>
1990 DATA 0,0,252,254,193,193,1,1,1,2,254,252,0,-1,-
     1                                                <073>
2000 DATA 0,0,1,3,2,2,2,2,1,1,0,-1,0,252,254,3,1,1,1
     ,1,2,4,0,-1,-1                                   <193>
2010 DATA 0,0,0,1,2,2,2,1,128,255,255,0,0,0,-1        <222>
2020 DATA 0,124,254,3,1,1,1,2,132,254,255,1,2,0,-1,-
     1                                                <204>
2030 DATA 0,0,1,3,2,2,2,2,1,0,0,-1                    <243>
2040 DATA 0,252,254,35,33,33,33,33,161,98,0,-1,-1     <190>
2050 DATA 0,15,63,114,130,131,131,128,96,32,0,-1      <233>
2060 DATA 0,255,255,0,0,0,0,0,0,0,-1,0,252,252,4,0
     ,0,0,0,0,0,0,-1                                  <004>
2070 DATA 0,0,0,1,2,2,2,1,1,1,1,1,0,-1                <095>
2080 DATA 0,124,254,131,1,1,1,130,132,191,127,128,0,
     -1                                               <216>
2090 DATA 0,56,52,4,4,4,4,8,16,240,224,0,0,-1         <163>
2100 DATA 0,128,255,255,1,2,2,2,1,1,0,0,0,0,-1        <124>
2110 DATA 0,0,255,255,0,0,0,0,0,254,255,1,2,0,-1,-1   <095>
2120 DATA 0,2,51,51,0,0,0,-1,0,0,254,255,1,2,0,-1,-1  <226>
2130 DATA 0,2,51,51,0,-1,0,0,255,255,0,-1,4,8,240,22
     4,0,-1                                           <245>
2140 DATA 0,128,255,255,0,0,0,3,3,0,0,0,-1            <168>
2150 DATA 0,0,255,255,32,88,76,134,3,1,2,0,-1,-1      <059>
2160 DATA 0,128,255,255,0,0,0,-1,0,0,254,255,1,2,0,-
     1,-1                                             <197>
2170 DATA 0,1,2,3,1,1,2,3,1,1,2,3,1,0,0,0,-1          <065>
2180 DATA 0,0,0,255,255,0,0,255,255,0,0,254,255,1,2,
     0,-1,-1                                          <160>
2190 DATA 0,1,2,3,1,1,2,3,1,0,0,0,-1                  <126>
2200 DATA 0,0,0,255,255,0,0,254,255,1,2,0,-1,-1       <131>
2210 DATA 0,0,1,1,2,2,2,2,3,1,0,0,-1                  <067>
2220 DATA 0,252,254,3,1,1,1,1,1,254,252,0,-1,-1       <110>
2230 DATA 0,1,2,3,1,0,1,2,2,2,3,1,0,0,-1              <109>
2240 DATA 0,0,0,255,255,132,2,1,1,1,3,254,252,0,-1    <211>
2250 DATA 0,0,0,252,252,4,0,0,0,0,0,0,0,0,-1          <199>
2260 DATA 0,0,0,1,2,2,2,1,1,1,1,1,0,-1                <029>
2270 DATA 0,124,254,131,1,1,1,130,132,191,127,128,0,
     -1                                               <156>
2280 DATA 0,0,0,0,0,0,0,0,0,252,252,4,0,-1            <239>
2290 DATA 0,1,2,3,1,2,3,1,1,0,-1,0,0,0,255,255,0,0,1
     28,0,0,-1,-1                                     <046>
2300 DATA 0,0,1,3,2,2,2,2,3,1,0,0,-1                  <161>
2310 DATA 0,198,247,49,49,49,49,49,179,159,12,0,-1,-
     1                                                <171>
2320 DATA 0,4,12,255,255,4,4,4,0,-1                   <003>
2330 DATA 0,0,0,254,255,1,2,0,0,-1,-1                 <247>
2340 DATA 0,2,3,3,0,0,0,2,3,3,0,0,0,-1                <179>
2350 DATA 0,0,254,255,1,1,1,2,254,255,1,2,0,-1,-1     <187>
2360 DATA 0,1,2,3,3,1,0,0,0,0,0,3,3,0,-1              <232>
2370 DATA 0,0,0,0,128,224,120,30,7,12,48,192,0,0,-1,
     -1                                               <035>
2380 DATA 0,2,3,3,0,0,3,3,0,0,3,3,0,-1                <109>
2390 DATA 0,0,254,255,1,2,254,255,1,2,254,252,0,-1,-
     1                                                <108>
2400 DATA 0,2,3,3,2,0,0,0,0,3,2,0,0,-1                <228>
2410 DATA 0,1,3,132,232,112,56,92,134,3,1,2,0,-1,-1   <142>
2420 DATA 0,1,2,3,3,1,0,0,0,0,0,3,3,0,-1              <036>
2430 DATA 0,0,0,0,128,224,120,30,7,12,48,192,0,0,-1   <172>
2440 DATA 0,0,0,24,28,12,48,192,0,0,0,0,0,0,-1        <036>
2450 DATA 0,3,3,2,2,2,2,2,3,3,2,0,-1                  <217>
2460 DATA 0,1,3,7,13,57,113,193,129,3,3,0,-1,-1       <001>
```


Listing Schreiberling. Verwenden Sie zur Eingabe den Checksummer V3.

Fit in Latein mit dem C 64

Mit »Cursus Latinus« können Sie sich Ihr eigenes elektronisches Lateinlehrbuch aufbauen und dieses lektionenweise studieren. Daß dabei die Besonderheiten der lateinischen Sprache berücksichtigt werden, macht das Programm so außergewöhnlich.

Grund zum Schreiben dieses Programms gaben mir einerseits meine schlechten Zensuren in Latein und andererseits mein Ärger darüber, daß alle mir bekannten Vokabellernprogramme für Lateinvokabeln gänzlich ungeeignet sind.

Große Probleme hatte ich anfangs mit der Abfrageroutine, aber ich habe das Programm so weit entwickelt, daß a) bei der zufälligen Abfrage der Vokabeln kein Wort doppelt abgefragt wird und b) die Bedeutung der Vokabeln in beliebiger Reihenfolge eingegeben werden können.

Das Programm ist an manchen Stellen recht langsam, allerdings empfehle ich nicht, es zu compilieren, denn dadurch würde der ohnehin knappe Speicher für Vokabeln noch kürzer.

Programmbeschreibung

Nach dem Starten des Programms wird dem Benutzer ein Menü mit fünf Punkten angeboten:

(a) Editieren (c) Diskette (e) Ende
(b) Abfragen (d) Drucken

(a) Editieren

Im Informationskasten (siehe Bild 1, Punkt a) sind die vier Hauptfunktionen dieses Programmteils dargestellt:

(1) Eingabe (2) Verbessern
(3) Löschen (4) Einfügen

Weitere Funktionen sind (eine Übersicht ist mit der Taste f8 abrufbar):

(f1)	Blättern 10 Einheiten vor
(f3)	Blättern 10 Einheiten zurück
(f5)	Blättern 1 Einheit vor
(f7)	Blättern 1 Einheit zurück
(+)	Index um 10 erhöhen
(—)	Index um 1 erhöhen
(←)	Hauptmenü

(1) Eingabe der Vokabeln

Bei der Eingabe muß zuerst die Wortart mit den Tasten < und > gewählt werden (die angewählte wird in weiß dargestellt), Symbole: Nom = Nomen Adj = Adjektiv Adv = Adverb Ver = Verb Dep = Deponens Sem = Semideponens

Konjunktionen sollten wie Adverbien behandelt werden.

Jetzt beginnt die eigentliche Eingabe: Der Cursor steht im Vokabel-Indexfeld (Bild 1). Der Vokabel-Index dient dazu, Wörter, die in alphabetischer Reihenfolge eingegeben wurden, nach Lektionen zu gliedern (zum Beispiel 27/05 bedeutet Lektion 27, Vers 5). Nach Drücken der RETURN-Taste wird nun, sofern man »Nomen« gewählt hat, der Reihenfolge nach das lateinische Wort, der Genitiv, der Genus und die drei Bedeutungen eingegeben. Hat man in der oberen Zeile einen Fehler gemacht und irrtümlicherweise »RETURN« gedrückt, kommt man mit SHIFT-RETURN wieder zurück. Die Eingabe ist erst mit der Taste f1 beendet, das Programm kehrt dann in das »Editier«-Menü zurück.

Hier noch einmal die Tastaturbelegung:

RETURN	nächste Eingabezeile
SHIFT-RETURN	wieder in die letzte Eingabezeile zurück
HOME	löscht aktuelle Eingabezeile
f1	Eingabe beendet
f3	löscht alle Eingabezeilen

(2) Verbessern

Um eine Vokabel zu verbessern, wählt man deren Nummer mit den Funktionstasten an.

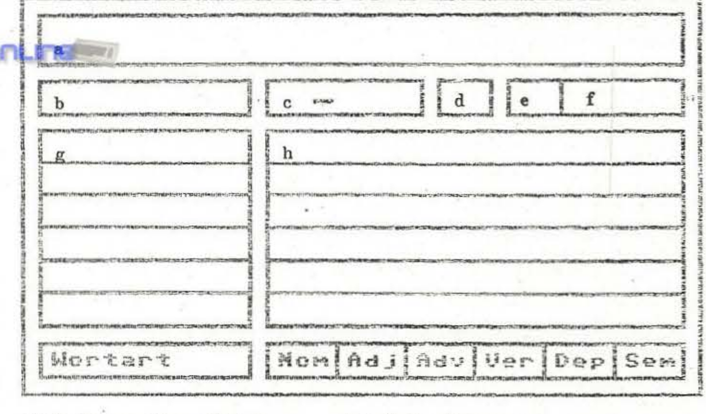
Zuerst muß wie bei der Eingabe die Wortart gewählt werden, dann werden die alten Zeilen angezeigt, die mit den aus der Eingabe bekannten Tasten verbessert werden können.

(3) Löschen

Wie auch beim Verbessern, sollte man vorher mit den Funktionstasten die zu löschende Vokabel-Nummer anwählen. Dann drückt man die Taste »3«.

(4) Einfügen

Bevor man diese Taste drückt, sollte man die einzufügende Vokabel-Nummer anwählen. Dies unterstützt die alphabetische Eingabe (falls man mal ein Wort übersehen hat, kann man diese Routine verwenden).



a) Informationsfeld b) Arbeitsgang
c) Wortart d) Nummer der Vokabel
e) Hauptindex f) Vokabel-Index
g) Bestimmungsstücke h) Eingabeteil

Bild 1. Die Aufteilung der Eingabemaske

»BLÄTTERN«

Mit den Funktionstasten kann man im Speicher herumblättern. Es ist die Eingabe von 100 Vokabeln vorgesehen (von 0 bis 99), die gerade angewählte Nummer wird im Kasten »d« (siehe Bild 1 und 2) angezeigt.

»INDEX«

Der Index dient dazu, eine formatierte Lektionensammlung auf Diskette zu erstellen (mit dem Index gibt man die einzelnen Lektionen an). Näheres unter »Speichern«.

Der Index wird in Kasten »e« (Bild 1) angezeigt und läßt sich mit den Tasten »+« und »—« ändern.

(b) Abfragen

Im Informationskasten der Abfrage (siehe Bild 3) sind folgende Möglichkeiten angezeigt.

(1) Abfragen in der Reihenfolge der Eingabe
(2) Zufällige Reihenfolge beim Abfragen
(f8) Informationen

Das »Abfragen in der Reihenfolge« dient eigentlich nur dazu, sich einen ersten Überblick über die Vokabeln zu verschaffen. Wenn man die Vokabeln in der Reihenfolge, in der man sie eingetippt hat, sieht, prägt man sie sich leichter ein. Wer diese Übung erfolgreich besteht, sollte jetzt die »Vokabeln zufällig« üben.

Nachdem alle Vokabeln abgefragt sind (gilt sowohl für »Reihenfolge« als auch für »Zufall«), wird das Ergebnis in absoluten Punkten und in Prozentzahlen angegeben (nach richtigen und falschen Antworten getrennt). Ich habe auf eine grafische Auswertung verzichtet, teils aus Speicherplatzmangel, teils weil ich meine, daß dies nur Spielerei ist. Jetzt hat man die Wahl, ob man alle Vokabeln nochmals wiederholen will, oder nur die nicht gewußten. Möchte man alle Wörter noch einmal durchgehen, so kann man sich wieder zwischen zufälliger oder alphabetischer Reihenfolge entscheiden.

Noch ein Wort zur Punktezählung beim Abfragen:

Für jede richtige Zeile erhält man einen Punkt. Hat man alle Zeilen richtig eingegeben, so gilt das Wort insgesamt als »richtig«. Ist auch nur eine falsch, so gilt es als »falsch«. Man kann übrigens die Bedeutungen in einer beliebigen Reihenfolge eingeben, sind alle richtig aber am falschen Platz, so zählen sie trotzdem als »richtig«.

Da es mir um ein reines Lernprogramm ging, habe ich auf alle überflüssigen und unnötigen Extras wie zum Beispiel einen brüllenden Löwen, hinaufkletternde Männchen und Abfragen nach Zeit verzichtet.

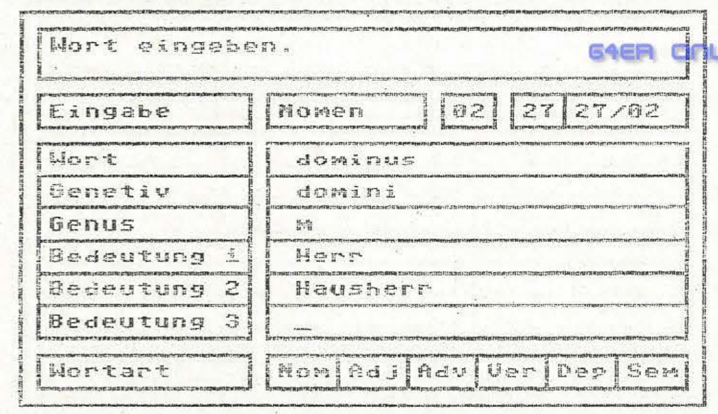
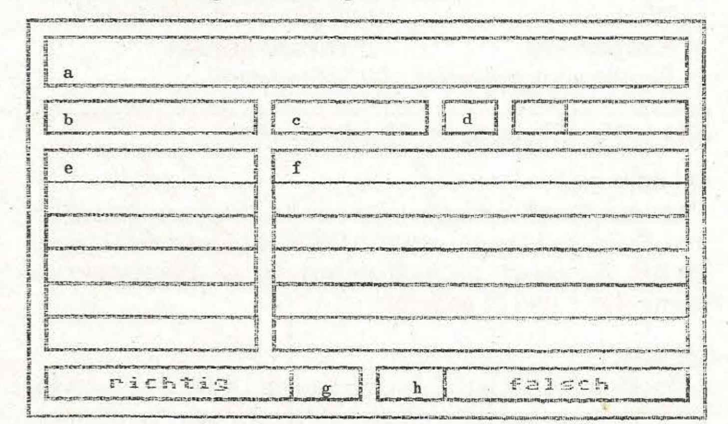


Bild 2. Eine ausgefüllte Eingabemaske



a) Informationskasten b) Arbeitsgang
c) Wortart d) Nummer der Vokabel
e) Bestimmungsstücke f) Eingabeteil
g) Punkte »richtig« h) Punkte »falsch«

Bild 3. Die Aufteilung der Abfragemaske

(c) Diskette

Dieser Programmteil unterscheidet zwischen folgenden Arbeitsgängen:
(1) Laden
(2) Speichern
(3) Laden nächster Index
und (←) zum Hauptmenü

Beim Laden und Speichern, das übrigens sequentiell vorgenommen wird, muß man zuerst den Filenamen eingeben und danach den Index einstellen. Wer dies schon bei der Eingabe getan hat, kann getrost »RETURN« drücken.

Variable	Verwendungszweck
T...$	Bildschirmmaske
WT$ (x,y)	Wortart mit Gliederung
MN$ (x)	Arbeitsgang
WO$ (x,y)	Vokabel (für y=0: Index der Vokabel)
WO (x)	Wortart der Vokabel (1 = Nomen 2 = Adjektiv 3 = Adverb 4 = Verb 5 = Deponens 6 = Semideponens)
DD$	Hilfsvariable zur Cursorpositionierung
FF$ (x)	Hilfsvariable Funktionstastendarstellung
LL$	Informationsfeld löschen
WA$ (x)	Wortart für Ausdruck
SP$	Trennzeile für Ausdruck
NR	Nummer der aktuellen Vokabel
NR$	String von NR
IN$	Index
Q1	Nächste Vokabel
ER,ES,ET,Q2	Hilfsvariablen mit unterschiedlichen Bedeutungen
I1,I2	Hilfsvariablen zur Index-Bearbeitung
ZF	Zufälligkeit bei der Abfrage (1 = Reihenfolge, 2 = zufällig)
R1	Gewußte Vokabelteile gesamt
R1$	String von R1
F1	Nicht gewußte Vokabelteile gesamt
F1$	String von F1
RQ	Gewußte Vokabelteile in der Vokabel
FO	Nicht gewußte Vokabelteile in der Vokabel
RF	Summe von RO und FO
MX,R2	Hilfsvariablen beim Abfragen
M(x)	Bedeutungen (0 = nicht gewußt 1 = gewußt)
UW$	Nicht gewußte Vokabeln
RD%	Position in ZF$
ZF$	ASCII-Werte der noch nicht abgefragten Vokabeln
FL$	Filename/Drucktitel
DS$	Diskerror
WX,LW	Hilfsvariablen zur Umformung von Zahlen in Strings
WX$	String der Zahl WX
PS	(56187) Farbspeicheradresse zur Wahl der Wortart
PA,AW	Hilfsvariablen zur Wahl der Wortart
ZZ	Gerade bearbeitete Eingabezeile
ZZ$ (x)	Eingabezeilen
ZZ (x)	Länge von ZZ$ (x)
BA	(53265) Adresse zum Ein- beziehungsweise Ausschalten des Bildschirms
SI	(54272) Basisadresse des Soundchips
TT	Übergabeparameter an Sound-Routinen
I,J,K	Laufvariablen
A$,AA$	Hilfsvariablen

Tabelle 1. Alle Variablen und deren Verwendungszweck

Der Filename setzt sich zusammen aus dem eingegebenen Namen, einem Punkt (».«) und dem Index. Das heißt, hat man einen Namen »Lektion« und den Index »27« gewählt, so heißt der Filename »Lektion.27«.

Einer Erklärung bedarf auch Punkt (3): Hierbei wird der Index um eins erhöht und dann das entsprechende Programm geladen. Ist kein Filename aus früheren Eingaben da, so bringt der Computer eine Fehlermeldung. Man sollte dann mit Punkt (1) weitermachen.

(d) Drucken

Nach dem Drücken der Taste (1) wird die Vokabelliste ausgedruckt, ein Abbruch ist mit »SHIFT LOCK« möglich.

DRUCKFORM
00 02/05 dominus; domini; m
(Nom.) Herr; Hausherr

Dabei bedeutet:

00 — Nummer	domini; m — Bestimmungsstücke
02/05 — Index	(Nom.) — Wortart
dominus — Wort	Herr; Hausherr — Bedeutungen

(e) Ende

Da das Programm und somit auch die Variablen (Tabelle 1 und 2) nicht gelöscht werden, ist eine Rückkehr ins Programm mit GOTO 830 möglich. Will man alle Vokabeln löschen, so sollte man diese Taste drücken und das Programm mit RUN starten.

Hinweise zum Abtippen

Die Druck-Routine ist für den MPS-801 geschrieben. Um sie an andere Drucker anzupassen, hier die Erklärungen:

4880 OPEN 1,4,7	schaltet Kleinschrift ein
4890 PRINT #1,CHR$(14);	Breitschrift
FL$	Drucktitel
CHR$(15)	Normalschrift

(Bernd Kraus/tr)

260		Variablenbelegung
830		Hauptmenü
1040	Editieren	
	1320	Eingabe
	1550	Verbessern
	1640	Löschen
	1780	Einfügen
	1870	Blättern vor
	1920	Blättern 10 vor
	1970	Blättern zurück
	2020	Blättern 10 zurück
	2070	Index ändern
	2160	Informationen
	2400	Fehlermeldung: Überlauf
2490	Abfragen	
	3280	Ergebnis
	3450	Fehlermeldung: Keine Vokabeln im Speicher
	3530	Informationen
3760	Diskette	
	3960	Laden
	4200	Speichern
	4440	Laden nächster Index
	4550	Diskerror
4670	Drucken	
5360	Unterprogramme	
	5400	Umrechnung Zahl in String der Länge 3
	5440	Umrechnung Zahl in String der Länge 2
	5480	Wahl der Wortart
	5590	Eingabe der Vokabel
	5920	Eingabe Filename
	6070	Laden/Speichern Eingabe Kontrolle
	6180	Info-Masken aufbauen
	6240	Sound

Tabelle 2. So ist das Programm aufgebaut

```
100 REM *************************              <153>
110 REM *                       *              <159>
120 REM *   - CURSUS LATINUS -  *              <130>
130 REM *                       *              <179>
140 REM *        MAERZ 1985     *              <231>
150 REM *                       *              <199>
160 REM *  BERND KRAUS          *              <044>
170 REM *  AM WALDSCHWIMMBAD 2  *              <108>
180 REM *  6909 WALLDORF/BD.    *              <156>
190 REM *  TEL.: 06227/2982     *              <022>
200 REM *                       *              <249>
210 REM *  FUER C=64 + C=1541   *              <210>
220 REM *  EVTL. MPS 801        *              <081>
230 REM *                       *              <023>
240 REM *************************              <037>
250 :                                          <226>
260 REM **                                     <055>
270 REM ** VARIABLEN                           <215>
280 REM **                                     <075>
290 :                                          <012>
300 PRINT"{CLR,WHITE,CTRL-N,CTRL-H}";:POKE
     53280,6:POKE 53281,6                      <210>
310 T1$="↑***********************************
    *******5"                                  <172>
320 T2$="7***********************************
    *******"                                   <008>
330 T3$="-↑**********************************
    ******5-"                                  <223>
340 T4$="-7**********************************
    ******X-"                                  <208>
350 T5$="--{36SPACE}--"                        <136>
360 T6$="-↑*************5↑**********5↑**5↑**R*
    ******5-"                                  <022>
370 T7$="--{11SHIFT-SPACE}--{8SHIFT-SPACE}
    --{2SHIFT-SPACE}--{2SHIFT-SPACE}-{6SHI
    FT-SPACE}--"                               <010>
380 T8$="-7*************X7**********X7**X7**F*
    ******X-"                                  <207>
390 T9$="-↑*************5↑********************
    ******5-"                                  <039>
400 TA$="--{11SHIFT-SPACE}--{23SHIFT-SPACE
    }--"                                       <095>
410 TB$="-7*************X7********************
    ******X-"                                  <243>
420 TC$="-↑****************R***5↑***R**********
    ******5-"                                  <101>
430 TD$="--{3SPACE,GREY 3}RICHTIG{WHITE,3S
    PACE}-{3SPACE}--{3SPACE}-{3SPACE,GREY
    3}FALSCH{WHITE,4SPACE}--"                  <016>
440 TE$="-7****************F***X7***F**********
    ******X-"                                  <203>
450 TF$="-↑*************5↑***R***R***R***R**
    *R***5-"                                   <244>
460 TG$="--{GREY 3}WORTART{4SPACE,WHITE}--
    {LIG.BLUE}NOM{WHITE}-{LIG.BLUE}ADJ{WHI
    TE}-{LIG.BLUE}ADV{WHITE}-{LIG.BLUE}VER
    {WHITE}-{LIG.BLUE}DEP{WHITE}-{LIG.BLUE
    }SEM{WHITE}--"                             <228>
470 TH$="-7*************X7***F***F***F***F**
    *F***X-"                                   <065>
480 TJ$="-↑**5↑*******************************
    ******5-"                                  <156>
490 TK$="--{2SHIFT-SPACE}--{32SHIFT-SPACE}
    --"                                        <075>
500 TL$="-7**X7******************************
    ******X-"                                  <230>
510 TM$="-{38SPACE}-"                          <234>
520 TN$="-Q*************WQ********************
    ******W-"                                  <160>
530 TU$="-↑*************5{13SPACE}↑***5↑****
    *5-"                                       <113>
540 TP$="--{2SHIFT-SPACE,GREY 3}RICHTIG{WH
    ITE,2SHIFT-SPACE}-{13SHIFT-SPACE}-{3SH
    IFT-SPACE}--{5SHIFT-SPACE}--"              <018>
550 TQ$="--{2SPACE,GREY 3}FALSCH{WHITE,3SP
    ACE}-{13SHIFT-SPACE}-{3SHIFT-SPACE}--{
    5SHIFT-SPACE}--"                           <007>
560 TR$="-7*************X{13SHIFT-SPACE}7***
    X7******X-"                                <090>
```

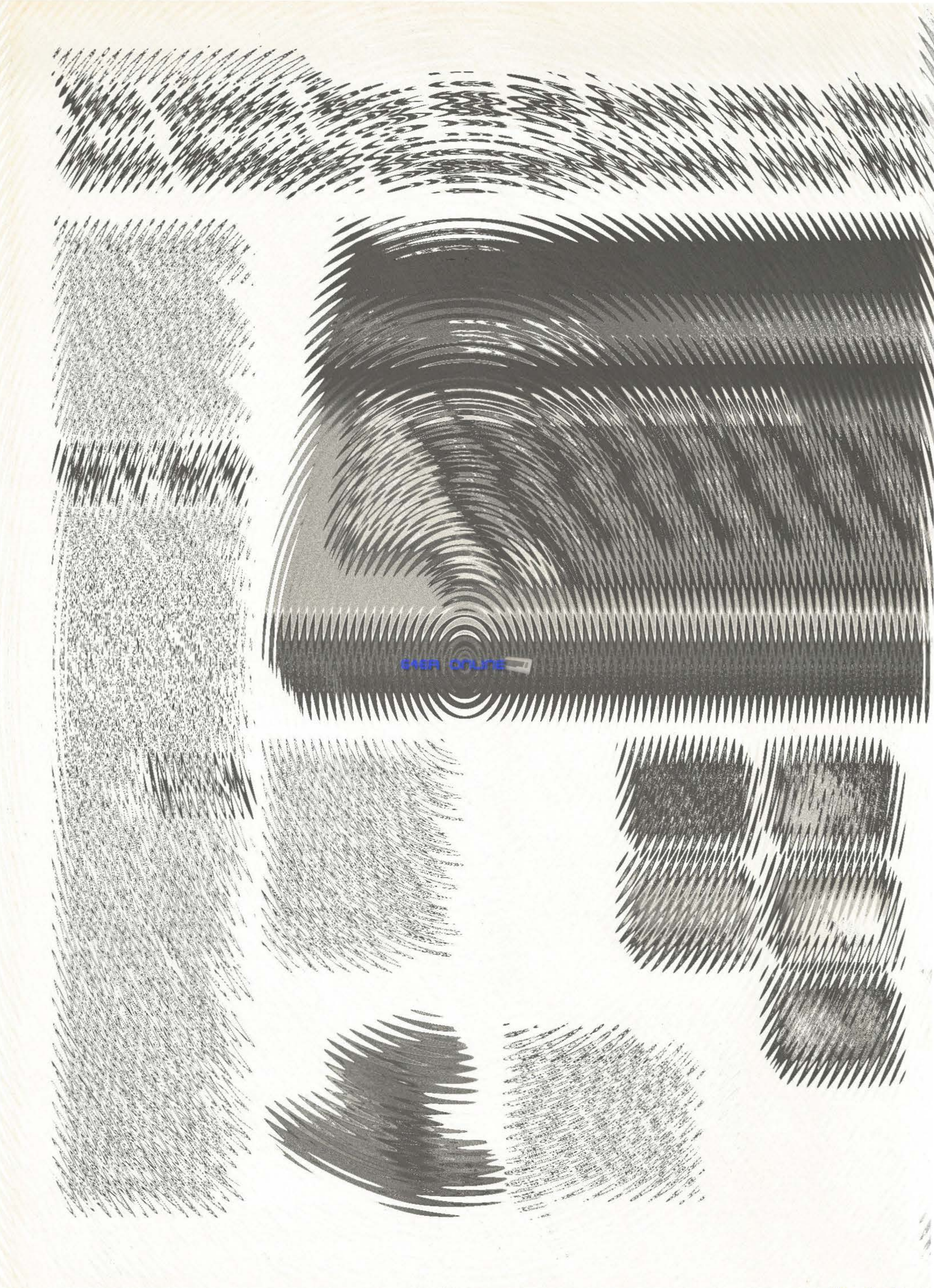
Listing »Cursus Latinus«. Beachten Sie bitte die Eingabehinweise auf Seite 54

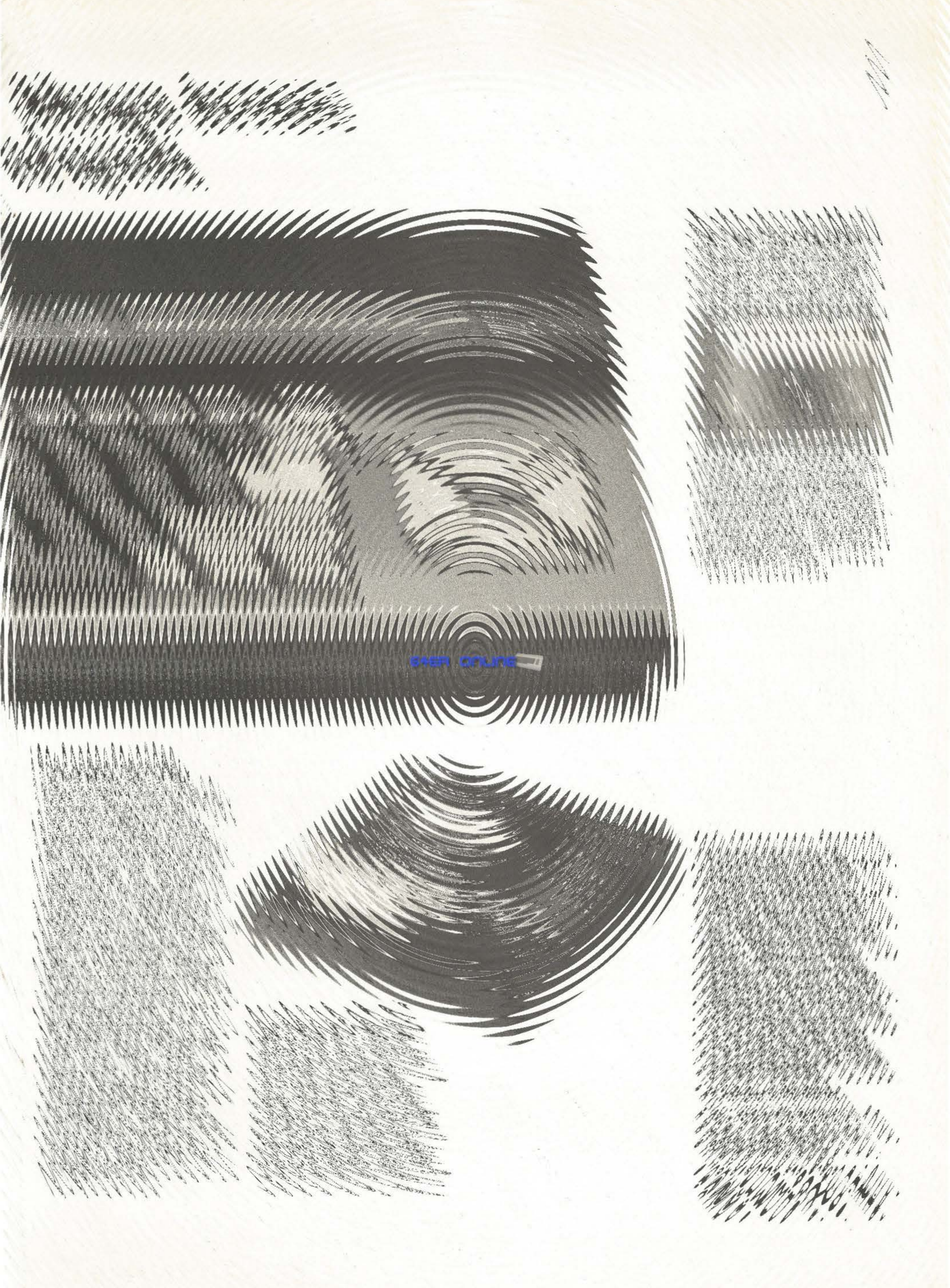
```
570 TT$="--{GREY 3}ABFRAGEN{20SPACE}ERGEBN
    IS{WHITE}--"                                 <124>
580 WT$(1,0)="NOMEN":WT$(1,1)="WORT":WT$(1
    ,2)="GENETIV":WT$(1,3)="GENUS"                <028>
590 WT$(1,4)="BEDEUTUNG 1":WT$(1,5)="BEDEU
    TUNG 2":WT$(1,6)="BEDEUTUNG 3"                <192>
600 WT$(2,0)="ADJEKTIV":WT$(2,1)="WORT":WT
    $(2,2)="FEMININ":WT$(2,3)="NEUTRUM"           <044>
610 WT$(3,0)="ADVERB":WT$(3,1)="WORT"             <054>
620 WT$(4,0)="VERB":WT$(4,1)="WORT":WT$(4,
    2)="STAMMFORM 1"                              <198>
630 WT$(4,3)="STAMMFORM 2":WT$(4,4)="STAMM
    FORM 3":WT$(4,5)=WT$(1,4)                     <174>
640 WT$(4,6)=WT$(1,5):WT$(5,0)="DEPONENS":
    WT$(5,1)="WORT":WT$(5,2)=WT$(4,2)             <065>
650 WT$(5,3)=WT$(4,3):WT$(6,0)="SEMIDEP.":
    FOR I=1 TO 3:WT$(6,I)=WT$(5,I):NEXT           <012>
660 FOR I=2 TO 6:IF I=4 THEN 680                  <184>
670 WT$(I,4)=WT$(1,4):WT$(I,5)=WT$(1,5):WT
    $(I,6)=WT$(1,6)                               <015>
680 NEXT                                          <182>
690 DIM MN$(12),WO$(99,6),WO(99)                  <228>
700 MN$(1)="HAUPTMENUE":MN$(2)="EDITIEREN"
    :MN$(3)="ABFRAGEN":MN$(4)="DISKETTE"          <226>
710 MN$(5)="DRUCKEN":MN$(6)="ENDE":MN$(7)=
    "EINGABE":MN$(8)="VERBESSERN"                 <245>
720 MN$(9)="LOESCHEN":MN$(10)="EINFUEGEN":
    MN$(11)="LADEN":MN$(12)="SPEICHERN"           <085>
730 DD$="{HOME,24DOWN}"                           <214>
740 FF$(0)="{2SPACE}":FF$(1)="{LIG.BLUE}F1
    {GREY 3}":FF$(2)="{LIG.BLUE}F3{GREY 3}
    ":FF$(3)="{LIG.BLUE}F5{GREY 3}":FF$(4)
    ="{LIG.BLUE}F7{GREY 3}"                       <037>
750 FF$(5)="{LIG.BLUE}F8{GREY 3}":NR=0:NR$
    ="00":IN$="00"                                <211>
760 LL$="{HOME,2DOWN,2RIGHT}":FOR I=1 TO 3
    6:LL$=LL$+" ":NEXT:LL$=LL$+"{4RIGHT}":
    FOR I=1 TO 36                                 <012>
770 LL$=LL$+" ":NEXT:BA=53265:SI=54272:POK
    E SI+24,15                                    <251>
780 WA$(1)="(NOM.)":WA$(2)="(ADJ.)":WA$(3)
    ="(ADV.)":WA$(4)="(VERB)"                     <166>
790 WA$(5)="(DEP.)":WA$(6)="(SEM.)"               <187>
800 FOR I=1 TO 80:SP$=SP$+"*":NEXT                <127>
810 :                                             <024>
820 :                                             <034>
830 REM **                                        <117>
840 REM ** HAUPTMENUE                             <184>
850 REM **                                        <137>
860 :                                             <074>
870 POKE BA,11                                    <089>
880 PRINT"{WHITE,CLR}";T1$T3$T5$T5$T4$T3$T
    5$T4$;:FOR I=1 TO 5:PRINT TJ$TK$TL$;:N
    EXT                                           <092>
890 PRINT TM$T2$;:POKE 2023,125                   <153>
900 PRINT"{GREY 3}";:FOR I=1 TO 6                 <095>
910 PRINT LEFT$(DD$,4+I*3);                       <012>
920 PRINT SPC(2);FF$(I-1);SPC(34-LEN(MN$(I
    )));MN$(I);:NEXT                              <250>
930 PRINT"{HOME,2DOWN,2RIGHT,WHITE,11SPACE
    }CURSUS{SHIFT-SPACE}LATINUS"                  <008>
940 PRINT"{UP,2RIGHT,GREY 3,6SPACE}VON BER
    ND KRAUS (C) 1985":POKE BA,27                 <032>
950 GET A$:IF A$=""THEN 950                       <127>
960 IF A$="{F1}"THEN 1040                         <223>
970 IF A$="{F3}"THEN 2490                         <222>
980 IF A$="{F5}"THEN 3760                         <084>
990 IF A$="{F7}"THEN 4670                         <235>
1000 IF A$="{F8}"THEN PRINT"{CLR}";:END           <092>
1010 GOTO 940                                     <048>
1020 :                                            <234>
1030 :                                            <246>
1040 REM **                                       <073>
1050 REM ** EDITIEREN                             <090>
1060 REM **                                       <093>
1070 :                                            <030>
1080 POKE BA,11                                   <045>
1090 PRINT"{WHITE,CLR}";T1$T3$T5$T5$T4$T6$
     T7$T8$T9$;:FOR I=1 TO 5:PRINT TA$TN$;
     :NEXT                                        <198>
1100 PRINT TA$TB$TF$TG$TH$T2$;:POKE 2023,1
     25:WX=NR:GOSUB 5440:NR$=WX$:PRINT"{GR
     EY 3}";                                      <010>
1110 PRINT LEFT$(DD$,7)"{2RIGHT}"MN$(2)"{2
     SPACE}"TAB(25)NR$TAB(29)IN$TAB(32)WO$
     (NR,0)                                       <227>
1120 PRINT LEFT$(DD$,7)SPC(15)WT$(WO(NR),0
     ):FOR I=1 TO 6                               <103>
1130 PRINT LEFT$(DD$,8+2*I)"{2RIGHT}"WT$(W
     O(NR),I):PRINT LEFT$(DD$,8+2*I)SPC(16
     );                                           <210>
1140 PRINT WO$(NR,I):NEXT                         <080>
1150 PRINT"{HOME,2DOWN,2RIGHT,LIG.BLUE,RVS
     ON,SPACE}1{SPACE,RVOFF,GREY 3,SPACE}E
     INGABE{3SPACE,LIG.BLUE,RVSON,SPACE}2{
     SPACE,RVOFF,GREY 3,SPACE}VERBESSERN{3
     SPACE,RVSON,LIG.BLUE,SPACE}F8{SPACE,R
     VOFF,GREY 3}"                                <236>
1160 PRINT"{HOME,3DOWN,2RIGHT,RVSON,LIG.BL
     UE,SPACE}3{SPACE,RVOFF,GREY 3,SPACE}L
     OESCHEN{2SPACE,LIG.BLUE,RVSON,SPACE}4
     {SPACE,GREY 3,RVOFF,SPACE}EINFUEGEN{4
     SPACE}INFO"                                  <249>
1170 POKE BA,27                                   <202>
1180 GET A$:IF A$=""THEN 1180                     <153>
1190 IF A$="1"THEN 1320                           <149>
1200 IF A$="2"THEN 1550                           <096>
1210 IF A$="3"THEN 1640                           <226>
1220 IF A$="4"THEN 1780                           <181>
1230 IF A$="{F1}"THEN 1920                        <023>
1240 IF A$="{F3}"THEN 2020                        <094>
1250 IF A$="{F5}"THEN 1870                        <116>
1260 IF A$="{F7}"THEN 1970                        <007>
1270 IF A$="+"OR A$="-"THEN 2070                  <230>
1280 IF A$="{F8}"THEN 2160                        <211>
1290 IF A$="←"THEN 830                            <008>
1300 GOTO 1180                                    <243>
1310 :                                            <016>
1320 REM ** EINGABE                               <128>
1330 IF Q1>99 THEN 2400                           <134>
1340 GOTO 1510                                    <122>
1350 PRINT LL$"{HOME,2DOWN,2RIGHT}WORTART
     MIT{SPACE,RVSON,LIG.BLUE,SPACE}<{SPAC
     E,RVOFF,GREY 3,SPACE}UND{SPACE,RVSON,
     LIG.BLUE,SPACE}>{SPACE,RVOFF,GREY 3,S
     PACE}WAEHLEN"                                <081>
1360 PRINT"{HOME,3DOWN,2RIGHT,RVSON,LIG.BL
     UE,SPACE}RETURN{SPACE,RVOFF,GREY 3}:
     ENDE DER EINGABE"                            <053>
1370 IF ET=1 THEN PRINT LEFT$(DD$,7)"{2RIG
     HT}"MN$(8):GOTO 1590                         <237>
1380 IF ET=2 THEN PRINT LEFT$(DD$,7)"{2RIG
     HT}"MN$(10)"{2SPACE}":GOTO 1400              <120>
1390 PRINT LEFT$(DD$,7)"{2RIGHT}"MN$(7)"{4
     SPACE}"                                      <143>
1400 GOSUB 5480                                   <183>
1410 PRINT LEFT$(DD$,7)SPC(15)WT$(AW,0)           <102>
1420 FOR I=1 TO 6:PRINT LEFT$(DD$,8+2*I)"{
     2RIGHT}"WT$(AW,I):NEXT:WO(Q1)=AW             <127>
1430 PRINT LL$"{HOME,2DOWN,2RIGHT}WORT EIN
     GEBEN"                                       <244>
1440 GOSUB 5590                                   <015>
1450 REM IFET=1THENRETURN                         <250>
1460 IF ET=0 THEN Q1=Q1+1                         <075>
1470 FOR I=1 TO 6:ZZ$(I)="":ZZ(I)=0:NEXT          <049>
1480 IF ET<>0 THEN RETURN                         <045>
1490 NR=Q1-1:GOTO 1040                            <006>
1500 :                                            <206>
1510 PRINT"{WHITE}"LEFT$(DD$,7)T7$;:FOR I=
     1 TO 6:PRINT LEFT$(DD$,8+2*I)TA$;:NEX
     T                                            <139>
1520 PRINT LEFT$(DD$,7)SPC(25);:WX=Q1:GOSU
     B 5440:PRINT"{GREY 3}"WX$;"{WHITE,2RI
     GHT,GREY 3}"IN$                              <034>
1530 GOTO 1350                                    <152>
1540 :                                            <248>
1550 REM ** VERBESSERN                            <115>
1560 IF Q1=0 THEN 1180                            <024>
1570 ET=1:Q2=Q1:Q1=NR:GOSUB 1320                  <166>
1580 Q1=Q2:NR=Q1-1:ET=0:GOTO 1040                 <063>
1590 ZZ$(0)=WO$(Q1,0):ZZ(0)=LEN(ZZ$(0)):PR
     INT LEFT$(DD$,7)SPC(32)ZZ$(0);               <005>
1600 FOR I=1 TO 6:ZZ$(I)=WO$(Q1,I):ZZ(I)=L
     EN(ZZ$(I)):PRINT LEFT$(DD$,8+2*I);           <111>
1610 PRINT SPC(16)ZZ$(I);:NEXT                    <149>
1620 GOTO 1400                                    <100>
1630 :                                            <082>
```

Listing »Cursus Latinus«. (Fortsetzung)

```
1640 REM ** LOESCHEN                             <171>
1650 IF Q1=0 THEN 1180                           <114>
1660 PRINT LEFT$(DD$,7)"{2RIGHT}"MN$(9)"{3
     SPACE}"                                     <175>
1670 PRINT LL$"{HOME,2DOWN,2RIGHT}DATENSAT
     Z WIRD GELOESCHT."                          <037>
1680 IF NR=Q1-1 THEN 1740                        <220>
1690 FOR I=NR TO Q1-2:FOR J=0 TO 6               <209>
1700 WO$(I,J)=WO$(I+1,J):NEXT:WO(I+1)=WO(I
     ):NEXT                                      <202>
1710 Q1=Q1-1:IF Q1=-1 THEN Q1=0                  <235>
1720 NR=NR-1:IF NR=-1 THEN NR=0                  <082>
1730 GOTO 1040                                   <018>
1740 FOR I=0 TO 6:WO$(NR,I)="":NEXT:WO(NR)
     =0                                          <137>
1750 Q1=Q1-1:NR=NR-1:IF NR=-1 THEN NR=0          <089>
1760 GOTO 1040                                   <048>
1770 :                                           <222>
1780 REM ** EINFUEGEN                            <174>
1790 IF Q1>99 THEN 2400                          <084>
1800 IF Q1=0 THEN 1180                           <010>
1810 PRINT LEFT$(DD$,7)"{2RIGHT}"MN$(10)"{
     2SPACE}"                                    <106>
1820 FOR I=Q1 TO NR+1 STEP-1:FOR J=0 TO 6        <122>
1830 WO$(I,J)=WO$(I-1,J):NEXT:WO(I)=WO(I-1
     ):NEXT                                      <082>
1840 Q2=Q1:Q1=NR:ET=2:GOSUB 1320:ET=0:NR=Q
     1:Q1=Q2:Q1=Q1+1                             <242>
1850 GOTO 1040                                   <140>
1860 :                                           <058>
1870 REM ** BLAETTERN VOR                        <108>
1880 IF Q1=0 THEN 1040                           <072>
1890 NR=NR+1:IF NR>Q1-1 OR NR>99 THEN NR=0
     :GOTO 1040                                  <231>
1900 GOTO 1040                                   <190>
1910 :                                           <108>
1920 REM ** BLAETTERN 10 VOR                     <177>
1930 IF Q1<10 THEN 1040                          <170>
1940 NR=NR+10:IF NR>Q1-1 OR NR>99 THEN NR=
     0:GOTO 1040                                 <032>
1950 GOTO 1040                                   <240>
1960 :                                           <158>
1970 REM ** BLAETTERN ZURUECK                    <097>
1980 IF Q1=0 THEN 1040                           <172>
1990 NR=NR-1:IF NR<0 THEN NR=Q1-1:GOTO 104
     0                                           <065>
2000 GOTO 1040                                   <034>
2010 :                                           <208>
2020 REM ** BLAETTERN 10 ZURUECK                 <093>
2030 IF Q1<10 THEN 1040                          <014>
2040 NR=NR-10:IF NR<0 THEN NR=Q1-1:GOTO 10
     40                                          <238>
2050 GOTO 1040                                   <086>
2060 :                                           <004>
2070 REM ** GESAMMTINDEX                         <083>
2080 I1=VAL(LEFT$(IN$,1)):I2=VAL(RIGHT$(IN
     $,1))                                       <163>
2090 IF A$="+"THEN I1=I1+1:IF I1>9 THEN I1
     =0                                          <184>
2100 IF A$="-"THEN I2=I2+1:IF I2>9 THEN I2
     =0                                          <029>
2110 WX=I1*10+I2:GOSUB 5440:IN$=WX$              <210>
2120 PRINT LEFT$(DD$,7)SPC(29)IN$                <114>
2130 IF ES=1 THEN GOTO 6130                      <185>
2140 GOTO 1180                                   <065>
2150 :                                           <094>
2160 REM ** INFO **                              <219>
2170 GOSUB 6180                                  <151>
2180 PRINT"{HOME,2DOWN,2RIGHT,GREY 3}*{2SP
     ACE}INFORMATIONEN ZUM PROGRAMMTEIL{2S
     PACE}*"                                     <001>
2190 PRINT"{UP,2RIGHT,WHITE,13SPACE}EDITIE
     REN{GREY 3}"                                <044>
2200 PRINT LEFT$(DD$,7)"{2RIGHT,SPACE,LIG.
     BLUE,SPACE}←{SPACE,GREY 3,SPACE}HAUPT
     MENUE"                                      <022>
2210 PRINT LEFT$(DD$,8)"{2RIGHT,LIG.BLUE,S
     PACE}F1{SPACE,GREY 3,SPACE}BLAETTERN
     10 EINHEITEN VOR"                           <073>
2220 PRINT LEFT$(DD$,9)"{2RIGHT,LIG.BLUE,S
     PACE}F3{SPACE,GREY 3,SPACE}BLAETTERN
     10 EINHEITEN ZURUECK"                       <232>
2230 PRINT LEFT$(DD$,10)"{2RIGHT,LIG.BLUE,
     SPACE}F5{SPACE,GREY 3,SPACE}BLAETTERN
     1 EINHEIT VOR"                              <147>
2240 PRINT LEFT$(DD$,11)"{2RIGHT,LIG.BLUE,
     SPACE}F7{SPACE,GREY 3,SPACE}BLAETTERN
     1 EINHEIT ZURUECK"                          <072>
2250 PRINT LEFT$(DD$,12)"{2RIGHT,SPACE,LIG
     .BLUE,SPACE}+{SPACE,GREY 3,SPACE}HAUP
     TINDEX UM 10 ERHOEHEN"                      <171>
2260 PRINT LEFT$(DD$,13)"{2RIGHT,SPACE,LIG
     .BLUE,SPACE}-{SPACE,GREY 3,SPACE}HAUP
     TINDEX UM 1 ERHOEHEN"                       <020>
2270 PRINT LEFT$(DD$,14)"{2RIGHT,SPACE,WHI
     TE}_{GREY 3}BEIM LOESCHEN/EINFUEGEN W
     IRD DIE{WHITE}_{GREY 3}"                    <236>
2280 PRINT LEFT$(DD$,15)"{2RIGHT,SPACE,WHI
     TE}_{GREY 3}ANGEWAEHLTE NUMMER GEL./E
     INGEF.{SHIFT-SPACE,WHITE}_{GREY 3}"         <207>
2290 PRINT LEFT$(DD$,17)"{2RIGHT,WHITE}EIN
     GABE/VERBESSERN:{GREY 3}"                   <237>
2300 PRINT LEFT$(DD$,18)"{2RIGHT,LIG.BLUE,
     SPACE}F1{SPACE,GREY 3,5SPACE}EINGABE
     BEENDET"                                    <110>
2310 PRINT LEFT$(DD$,19)"{2RIGHT,LIG.BLUE,
     SPACE}F3{SPACE,GREY 3,5SPACE}ALLE ZEI
     LEN LOESCHEN"                               <152>
2320 PRINT LEFT$(DD$,20)"{2RIGHT,LIG.BLUE,
     SPACE}HOME{SPACE,GREY 3,3SPACE}ZEILE
     LOESCHEN"                                   <087>
2330 PRINT LEFT$(DD$,21)"{2RIGHT,LIG.BLUE,
     SPACE}RET{SPACE,GREY 3,4SPACE}ZEILE W
     EITER"                                      <053>
2340 PRINT LEFT$(DD$,22)"{2RIGHT,LIG.BLUE,
     SPACE}SF-RET{SPACE,GREY 3,SPACE}ZEILE
      HOCH"                                      <152>
2350 PRINT LEFT$(DD$,23)"{2RIGHT,20SPACE}<
     TASTE DRUECKEN>"                            <156>
2360 POKE BA,27                                  <122>
2370 GET A$:IF A$=""THEN 2370                    <104>
2380 GOTO 1040                                   <162>
2390 :                                           <080>
2400 REM ** UEBERLAUF                            <000>
2410 PRINT LL$"{HOME,2DOWN,2RIGHT}ZU VIELE
      DATEN IM SPEICHER!"                        <141>
2420 PRINT"{UP,2RIGHT,SPACE}< TASTE DRUECK
     EN >"                                       <210>
2430 TT=12:GOSUB 6250                            <234>
2440 GET A$:IF A$=""THEN 2440                    <110>
2450 POKE SI+4,0                                 <216>
2460 GOTO 1040                                   <242>
2470 :                                           <160>
2480 :                                           <170>
2490 REM **                                      <253>
2500 REM ** ABFRAGEN                             <225>
2510 REM **                                      <017>
2520 :                                           <210>
2530 POKE BA,11:ZF=0:R0=0:F0=0:R1=0:F1=0         <096>
2540 PRINT"{WHITE,CLR}";T1$T3$T5$T5$T4$T6$
     T7$T8$T9$;:FOR I=I=1 TO 4:PRINT TA$TN
     $;:NEXT                                     <255>
2550 PRINT TA$TB$TC$TD$TE$T2$;:POKE 2023,1
     25                                          <074>
2560 PRINT LEFT$(DD$,7)"{GREY 3,2RIGHT}"MN
     $(3)"{3SPACE}"                              <155>
2570 POKE BA,27                                  <078>
2580 IF Q1=0 THEN 3450                           <024>
2590 IF ER<>1 THEN UW$=""                        <239>
2600 IF ER=1 THEN ER=0:ZF=2:ZF$=UW$:UW$=""
     :GOTO 2700                                  <194>
2610 PRINT"{HOME,2DOWN,2RIGHT,RVSON,LIG.BL
     UE,SPACE}1{SPACE,RVOFF,GREY 3,SPACE}A
     BFRAGEN IN DER REIHENFOLGE{SPACE,RVSO
     N,LIG.BLUE,SPACE}F8 "                       <132>
2620 PRINT"{UP,2RIGHT,RVSON,SPACE}2{SPACE,
     RVOFF,GREY 3,SPACE}ZUFAELLIGE REIHENF
     OLGE{6SPACE}INFO"                           <170>
2630 GET A$:IF A$=""THEN 2630                    <045>
2640 IF A$="1"THEN ZF=1:GOTO 2690                <006>
2650 IF A$="2"THEN ZF=2:GOTO 2690                <177>
2660 IF A$="←"THEN 830                           <108>
2670 IF A$="{F8}"THEN 3540                       <079>
2680 GOTO 2630                                   <024>
2690 ZF$="":FOR I=0 TO Q1-1:ZF$=ZF$+CHR$(I
     +32):NEXT                                   <072>
```

Listing »Cursus Latinus«. (Fortsetzung) Weiter auf Seite 64





```
2700 FOR HS=0 TO LEN(ZF$)-1:PRINT LL$:RF=0
     :R0=0:F0=0                                <167>
2710 IF ZF=1 THEN NR=HS:GOTO 2760              <219>
2720 RD%=RND(1)*(LEN(ZF$)-1)+1                 <025>
2730 NR=ASC(MID$(ZF$,RD%))-32                  <165>
2740 IF RD%=1 THEN ZF$=RIGHT$(ZF$,LEN(ZF$)
     -RD%):GOTO 2760                           <093>
2750 ZF$=LEFT$(ZF$,RD%-1)+RIGHT$(ZF$,LEN(Z
     F$)-RD%)                                  <187>
2760 PRINT LEFT$(DD$,7)SPC(15)"{8SPACE}":W
     X=NR:GOSUB 5440:NR$=WX$                   <238>
2770 PRINT LEFT$(DD$,7)SPC(15)WT$(WO(NR),0
     ):PRINT LEFT$(DD$,7)SPC(25)NR$            <220>
2780 FOR J=1 TO 6:AA$=LEFT$(DD$,8+2*J):PRI
     NT AA$"{2RIGHT,11SPACE}"                  <136>
2790 PRINT AA$"{2RIGHT}"WT$(WO(NR),J);         <131>
2800 PRINT AA$SPC(15)"{21SPACE}"               <109>
2810 NEXT                                      <024>
2820 PRINT LEFT$(DD$,10)SPC(16)"{WHITE}"WO
     $(NR,1)"{GREY 3}"                         <038>
2830 PRINT LEFT$(DD$,23)"{2RIGHT,3SPACE}RI
     CHTIG{3SPACE}"TAB(25)"{3SPACE}FALSCH{
     4SPACE}"                                  <050>
2840 ES=2:GOSUB 5590:ES=0                      <078>
2850 IF WO(NR)=3 THEN 2930                     <154>
2860 ER=0:IF WO(NR)=2 THEN ER=1                <165>
2870 FOR J=2 TO 3+ER                           <084>
2880 IF WO$(NR,J)=""THEN 2920                  <186>
2890 IF ZZ$(J)=WO$(NR,J)THEN R0=R0+1:GOTO
     2910                                      <207>
2900 F0=F0+1                                   <137>
2910 RF=RF+1                                   <123>
2920 NEXT                                      <136>
2930 IF ER=1 THEN 3200                         <082>
2940 MX=3:FOR J=4 TO 6                         <111>
2950 M(J)=0:W$=WO$(NR,J):IF W$=""THEN MX=M
     X-1:GOTO 2980                             <253>
2960 IF W$=ZZ$(4)OR W$=ZZ$(5)OR W$=ZZ$(6)T
     HEN M(J)=1                                <142>
2970 RF=RF+1                                   <183>
2980 NEXT                                      <196>
2990 R2=M(4)+M(5)+M(6):F0=F0+MX-R2:R0=R0+R
     2                                         <208>
3000 IF RF<>R0 THEN UW$=UW$+CHR$(NR+32)        <241>
3010 IF R0=0 THEN PRINT LL$"{HOME,2DOWN,2R
     IGHT}WORT NICHT GEWUSST.":GOTO 3050       <100>
3020 IF RF<>R0 THEN PRINT LL$"{HOME,2DOWN,
     2RIGHT}WORT UNVOLLSTAENDIG GEWUSST.":
     GOTO 3050                                 <122>
3030 PRINT LL$"{HOME,2DOWN,2RIGHT}WORT GEW
     USST.":PRINT LEFT$(DD$,23)"{2RIGHT,RV
     SON,3SPACE}RICHTIG{3SPACE,RVOFF}";        <223>
3040 TT=35:GOSUB 6250:R1=R1+R0:GOTO 3120       <142>
3050 PRINT LEFT$(DD$,23)TAB(25)"{RVSON,3SP
     ACE}FALSCH{4SPACE,RVOFF}";:GOSUB 6270     <242>
3060 FOR K=1 TO 300:NEXT                       <009>
3070 PRINT LL$"{HOME,2DOWN,2RIGHT}RICHTIG
     LAUTET ES:"                               <006>
3080 FOR J=2 TO 6:PRINT LEFT$(DD$,8+2*J)SP
     C(15)"{21SPACE}"                          <205>
3090 PRINT LEFT$(DD$,8+2*J)SPC(16)WO$(NR,J
     )                                         <074>
3100 NEXT                                      <062>
3110 R1=R1+R0:F1=F1+F0                         <122>
3120 WX=R1:GOSUB 5400:R1$=WX$:WX=F1:GOSUB
     5400:F1$=WX$                              <174>
3130 PRINT LEFT$(DD$,23)TAB(16)"{WHITE}"R1
     $TAB(21)F1$"{GREY 3}"                     <209>
3140 PRINT"{HOME,3DOWN,2RIGHT,SPACE}< TAST
     E DRUECKEN >"                             <018>
3150 GET A$:IF A$=""THEN 3150                  <025>
3160 POKE SI+4,0                               <164>
3170 IF A$="←"THEN FOR I=0 TO 6:ZZ$(I)="":
     ZZ(I)=0:NEXT:GOTO 830                     <230>
3180 FOR J=2 TO 6:ZZ$(J)="":ZZ(J)=0:NEXT       <117>
3190 NEXT:GOTO 3280                            <063>
3200 MX=2:FOR J=5 TO 6                         <089>
3210 M(J)=0::W$=WO$(NR,J):IF W$=""THEN MX=
     MX-1:GOTO 3240                            <029>
3220 IF W$=ZZ$(5)OR W$=ZZ$(6)THEN M(J)=1       <213>
3230 RF=RF+1                                   <189>
3240 NEXT                                      <202>
3250 R2=M(5)+M(6):F0=F0+MX-R2:R0=R0+R2         <001>
3260 GOTO 3000                                 <168>
3270 :                                         <198>
3280 REM ** ERGEBNIS                           <121>
3290 POKE BA,11                                <223>
3300 PRINT"{WHITE,CLR}"T1$T3$T5$T5$T4$T3$T
     T$T4$TM$TM$TM$TU$TP$TR$TU$TQ$TR$;         <248>
3310 PRINT TM$TM$TM$TM$T3$T5$T4$T2$;:POKE
     2023,125                                  <184>
3320 PRINT"{HOME,2DOWN,2RIGHT,LIG.BLUE,RVS
     ON,SPACE}1{SPACE,RVOFF,GREY 3,SPACE}A
     LLE NICHT GEW. VOK. WIEDERHOLEN"          <193>
3330 PRINT"{UP,2RIGHT,RVSON,LIG.BLUE,SPACE
     }2{SPACE,RVOFF,GREY 3,SPACE}ALLE VOK.
      WIEDERHOLEN{2SPACE,RVSON,LIG.BLUE,SP
     ACE}←{SPACE,RVOFF,GREY 3,SPACE}MENUE"     <140>
3340 RF=R1+F1:RR=INT(R1/RF*100+.5):FF=100-
     RR                                        <142>
3350 WX=RR:GOSUB 5400:RR$=WX$:WX=FF:GOSUB
     5400:FF$=WX$                              <109>
3360 PRINT LEFT$(DD$,13)SPC(28)R1$"{2RIGHT
     }"RR$" %"                                 <100>
3370 PRINT LEFT$(DD$,16)SPC(28)F1$"{2RIGHT
     }"FF$"{SHIFT-SPACE}%"                     <055>
3380 POKE BA,27                                <126>
3390 GET A$:IF A$=""THEN 3390                  <141>
3400 IF A$="←"THEN 830                         <086>
3410 IF A$="1"THEN IF F1<>0 THEN ER=1:GOTO
     2530                                      <121>
3420 IF A$="2"THEN 2530                        <002>
3430 GOTO 3390                                 <165>
3440 :                                         <114>
3450 REM ** KEINE VOKABELN                     <196>
3460 PRINT LL$"{HOME,2DOWN,2RIGHT}KEINE VO
     KABELN IM SPEICHER."                      <136>
3470 PRINT"{UP,3RIGHT}< TASTE DRUECKEN >"      <094>
3480 TT=12:GOSUB 6250                          <012>
3490 GET A$:IF A$=""THEN 3490                  <050>
3500 POKE SI+4,0                               <250>
3510 GOTO 830                                  <240>
3520 :                                         <194>
3530 REM ** INFO                               <067>
3540 GOSUB 6180                                <251>
3550 PRINT"{HOME,2DOWN,2RIGHT,GREY 3}*{2SP
     ACE}INFORMATIONEN ZUM PROGRAMMTEIL{2S
     PACE}*"                                   <101>
3560 PRINT"{UP,2RIGHT,WHITE,14SPACE}ABFRAG
     EN{GREY 3}"                               <192>
3570 PRINT LEFT$(DD$,8)"{2RIGHT,WHITE}EING
     ABE{GREY 3}"                              <156>
3580 PRINT LEFT$(DD$,9)"{2RIGHT,LIG.BLUE,S
     PACE}F1{SPACE,GREY 3,5SPACE}EINGABE B
     EENDET"                                   <150>
3590 PRINT LEFT$(DD$,10)"{2RIGHT,LIG.BLUE,
     SPACE}F3{SPACE,GREY 3,5SPACE}ALLE ZEI
     LEN LOESCHEN"                             <126>
3600 PRINT LEFT$(DD$,11)"{2RIGHT,LIG.BLUE,
     SPACE}HOME{SPACE,GREY 3,3SPACE}ZEILE
     LOESCHEN"                                 <099>
3610 PRINT LEFT$(DD$,12)"{2RIGHT,LIG.BLUE,
     SPACE}RET{SPACE,GREY 3,4SPACE}ZEILE W
     EITER"                                    <065>
3620 PRINT LEFT$(DD$,13)"{2RIGHT,LIG.BLUE,
     SPACE}SF-RET{SPACE,GREY 3,SPACE}ZEILE
      HOCH"                                    <164>
3630 PRINT LEFT$(DD$,14)"{2RIGHT,LIG.BLUE,
     SPACE}←{SPACE,GREY 3,6SPACE}ABFRAGE B
     EENDEN"                                   <129>
3640 PRINT LEFT$(DD$,16)"{2RIGHT,WHITE}AUS
     WAHL{GREY 3}"                             <251>
3650 PRINT LEFT$(DD$,17)"{2RIGHT,LIG.BLUE,
     SPACE}1{SPACE,GREY 3,SPACE}FRAEGT ALL
     E VOKABELN DER REIHEN-"                   <150>
3660 PRINT LEFT$(DD$,18)"{2RIGHT,4SPACE}FO
     LGE NACH AB."                             <227>
3670 PRINT LEFT$(DD$,19)"{2RIGHT,LIG.BLUE,
     SPACE}2{SPACE,GREY 3,SPACE}WAEHLT EIN
     E ZUFAELLIGE REIHEN-"                     <197>
3680 PRINT LEFT$(DD$,20)"{2RIGHT,4SPACE}FO
     LGE."                                     <113>
3690 PRINT LEFT$(DD$,21)"{2RIGHT,LIG.BLUE,
     SPACE}←{SPACE,GREY 3,SPACE}HAUPTMENUE
     "                                         <180>
3700 PRINT LEFT$(DD$,23)"{2RIGHT,20SPACE}<
     TASTE DRUECKEN>"                          <236>
3710 POKE BA,27                                <202>
3720 GET A$:IF A$=""THEN 3720                  <087>
3730 GOTO 2490                                 <219>
```

```
3740 :                                    <160>
3750 :                                    <170>
3760 REM **                               <253>
3770 REM ** DISKETTE                      <166>
3780 REM **                               <017>
3790 :                                    <210>
3800 POKE BA,11                           <225>
3810 PRINT"{WHITE,CLR}"T1$T3$T5$T5$T4$T6$T
     7$T8$T9$TA$TB$;:FOR I=1 TO 10:PRINT T
     M$;:NEXT                             <114>
3820 PRINT T3$T5$T4$T2$;:POKE 2023,125    <222>
3830 PRINT LEFT$(DD$,7)"{2RIGHT,GREY 3}"MN
     $(4)"{3SPACE}"TAB(29)IN$             <250>
3840 PRINT LEFT$(DD$,10)"{2RIGHT}FILENAME:
     {5RIGHT}"FL$                         <051>
3850 PRINT LEFT$(DD$,23)"{2RIGHT}00, OK,00
     ,00"                                 <235>
3860 PRINT LL$"{HOME,2DOWN,2RIGHT,RVSON,LI
     G.BLUE,SPACE}1{SPACE,GREY 3,RVOFF,SPA
     CE}LADEN{3SPACE,RVSON,LIG.BLUE,SPACE}
     2{SPACE,GREY 3,RVOFF,SPACE}SPEICHERN{
     7SPACE,RVSON,LIG.BLUE,SPACE}←{SPACE,G
     REY 3,RVOFF}"                        <027>
3870 PRINT"{HOME,3DOWN,2RIGHT,RVSON,LIG.BL
     UE,SPACE}3{SPACE,GREY 3,RVOFF,SPACE}L
     ADEN NAECHSTER INDEX{6SPACE}MENUE"   <005>
3880 POKE BA,27                           <118>
3890 GET A$:IF A$=""THEN 3890             <199>
3900 IF A$="←"THEN 830                    <078>
3910 IF A$="1"THEN 3960                   <195>
3920 IF A$="2"THEN 4200                   <186>
3930 IF A$="3"THEN 4440                   <148>
3940 GOTO 3890                            <247>
3950 :                                    <116>
3960 REM ** LADEN                         <163>
3970 GOSUB 6070                           <124>
3980 IF FL$=""THEN 3800                   <022>
3990 OPEN 1,8,2,FL$+"."+IN$+",S,R"        <204>
4000 OPEN 2,8,15                          <110>
4010 INPUT#1,Q1                           <155>
4020 IF ST=66 THEN 4550                   <108>
4030 FOR I=0 TO Q1-1                      <122>
4040 FOR J=0 TO 6                         <179>
4050 INPUT#1,WO$(I,J)                     <106>
4060 IF WO$(I,J)="₸"THEN WO$(I,J)=""      <213>
4070 IF ST=66 THEN 4550                   <158>
4080 NEXT                                 <024>
4090 INPUT#1,WO(I)                        <225>
4100 IF ST=66 THEN 4550                   <190>
4110 NEXT                                 <056>
4120 CLOSE 2:CLOSE 1                      <078>
4130 PRINT LL$"{HOME,2DOWN,2RIGHT}DATEN SI
     ND GELADEN."                         <193>
4140 TT=45:GOSUB 6250                     <199>
4150 PRINT"{UP,2RIGHT,SPACE}< TASTE DRUECK
     EN >"                                <162>
4160 GET A$:IF A$=""THEN 4160             <180>
4170 POKE SI+4,0                          <158>
4180 GOTO 830                             <148>
4190 :                                    <102>
4200 REM ** SPEICHERN                     <225>
4210 IF Q1=0 THEN 3450                    <130>
4220 GOSUB 6070:IF FL$=""THEN 3800        <172>
4230 OPEN 1,8,2,FL$+"."+IN$+",S,W"        <200>
4240 OPEN 2,8,15                          <096>
4250 PRINT#1,Q1                           <221>
4260 IF ST=66 THEN 4550                   <094>
4270 FOR I=0 TO Q1-1                      <108>
4280 FOR J=0 TO 6                         <165>
4290 IF WO$(I,J)=""THEN PRINT#1,"₸":GOTO 4
     310                                  <004>
4300 PRINT#1,WO$(I,J)                     <182>
4310 IF ST=66 THEN 4550                   <144>
4320 NEXT                                 <010>
4330 PRINT#1,WO(I)                        <035>
4340 IF ST=66 THEN 4550                   <174>
4350 NEXT                                 <040>
4360 CLOSE 2:CLOSE 1                      <064>
4370 PRINT LL$"{HOME,2DOWN,2RIGHT}DATEN SI
     ND ABGESPEICHERT."                   <002>
4380 PRINT"{UP,2RIGHT,SPACE}< TASTE DRUECK
     EN >"                                <138>
4390 TT=45:GOSUB 6250                     <195>
4400 GET A$:IF A$=""THEN 4400             <100>
4410 POKE SI+4,0                          <144>
4420 GOTO 830                             <134>
4430 :                                    <088>
4440 REM ** LADEN NAECHSTER INDEX         <155>
4450 IF FL$<>""THEN 4510                  <026>
4460 PRINT LL$"{HOME,2DOWN,2RIGHT}KEINE GR
     UNDDATEI VORHANDEN."                 <039>
4470 PRINT"{UP,3RIGHT}< TASTE DRUECKEN >" <078>
4480 TT=12:GOSUB 6250                     <252>
4490 GET A$:IF A$=""THEN 4490             <066>
4500 POKE SI+4,0:GOTO 3860                <114>
4510 IN=VAL(IN$):IN=IN+1:IF IN>99 THEN IN=
     0                                    <253>
4520 WX=IN:GOSUB 5440:IN$=WX$             <002>
4530 PRINT LEFT$(DD$,7)SPC(29)IN$:GOTO 399
     0                                    <226>
4540 :                                    <198>
4550 REM ** DISKERROR                     <138>
4560 DS$=""                               <206>
4570 GET#2,A$:DS$=DS$+A$:IF ST<>64 THEN 45
     70                                   <003>
4580 PRINT LEFT$(DD$,23)"{2RIGHT}";:FOR I=
     1 TO 36:PRINT" ";:NEXT               <044>
4590 PRINT LEFT$(DD$,23)"{2RIGHT}"DS$;    <254>
4600 CLOSE 2:CLOSE 1                      <048>
4610 PRINT LL$"{HOME,2DOWN,2RIGHT}DISKERRO
     R"                                   <217>
4620 PRINT"{UP,2RIGHT,SHIFT-SPACE}< TASTE
     DRUECKEN >"                          <168>
4630 GET A$:IF A$=""THEN 4630             <077>
4640 GOTO 3800                            <187>
4650 :                                    <054>
4660 :                                    <064>
4670 REM **                               <147>
4680 REM ** DRUCKEN                       <195>
4690 REM **                               <167>
4700 :                                    <104>
4710 POKE BA,11                           <119>
4720 PRINT"{WHITE,CLR}"T1$T3$T5$T5$T4$T6$T
     7$T8$T9$TA$TB$;:FOR I=1 TO 10:PRINT T
     M$;:NEXT                             <008>
4730 PRINT T3$T5$T4$T2$;:POKE 2023,125    <116>
4740 PRINT LEFT$(DD$,7)"{GREY 3,2RIGHT}"MN
     $(5)"{4SPACE}";                      <138>
4750 PRINT LEFT$(DD$,10)"{2RIGHT}DRUCKTITE
     L:"                                  <096>
4760 PRINT"{HOME,2DOWN,2RIGHT,LIG.BLUE,RVS
     ON,SPACE}1{SPACE,RVOFF,GREY 3,SPACE}D
     RUCKEN"                              <203>
4770 PRINT"{UP,2RIGHT,LIG.BLUE,RVSON,SPACE
     }←{SPACE,RVOFF,GREY 3,SPACE}MENUE"   <205>
4780 POKE BA,27                           <000>
4790 IF Q1=0 THEN 3450                    <202>
4800 GET A$:IF A$=""THEN 4800             <247>
4810 IF A$="←"THEN 830                    <226>
4820 IF A$="1"THEN 4840                   <051>
4830 GOTO 4800                            <094>
4840 PRINT LL$"{HOME,2DOWN,2RIGHT,GREY 3}D
     RUCKTITEL EINGEBEN."                 <037>
4850 ES=7:GOSUB 5920:ES=0                 <194>
4860 IF FL$=""THEN 830                    <114>
4870 PRINT LL$"{HOME,2DOWN,2RIGHT,GREY 3}A
     BBRUCH MIT{SPACE,LIG.BLUE,RVSON,SPACE
     }SHIFT LOCK{SPACE,RVOFF,GREY 3}"     <053>
4880 OPEN 1,4,7                           <043>
4890 PRINT#1,CHR$(14)FL$CHR$(15)          <235>
4900 PRINT#1,SP$:PRINT#1                  <182>
4910 I=-1                                 <238>
4920 I=I+1:IF I=Q1 THEN 5090              <224>
4930 WX=I:GOSUB 5440:II$=WX$              <047>
4940 PRINT#1,II$;" ";WO$(I,0)+LEFT$("{6SPA
     CE}",6-LEN(WO$(I,0)));" ";           <219>
4950 PRINT#1,WO$(I,1);                    <228>
4960 IF WO(I)=4 THEN 5110                 <243>
4970 IF WO(I)=5 OR WO(I)=6 THEN 5180      <089>
4980 IF WO$(I,2)<>""THEN PRINT#1,"; "WO$(I
     ,2);                                 <113>
4990 IF WO$(I,3)<>""THEN PRINT#1,"; "WO$(I
     ,3);                                 <141>
5000 PRINT#1                              <153>
5010 PRINT#1,"{2SPACE}";WA$(WO(I));"{2SPAC
     E}";                                 <133>
```

Listing »Cursus Latinus«. (Fortsetzung)

```
5020 PRINT#1,WO$(I,4);                          <066>
5030 IF WO$(I,5)<>""THEN PRINT#1,"; "WO$(I
     ,5);                                       <217>
5040 IF WO$(I,6)<>""THEN PRINT#1,"; "WO$(I
     ,6);                                       <245>
5050 PRINT#1                                    <203>
5060 PRINT#1                                    <213>
5070 IF PEEK(653)=0 THEN 4920                   <255>
5080 GOTO 5230                                  <098>
5090 PRINT#1:PRINT#1:CLOSE 1                    <020>
5100 GOTO 5280                                  <023>
5110 PRINT#1                                    <007>
5120 PRINT#1,"{2SPACE}"WA$(4)"{2SPACE}";        <017>
5130 PRINT#1,WO$(I,2);"; "WO$(I,3)"; "WO$(
     I,4);                                      <150>
5140 PRINT#1                                    <039>
5150 PRINT#1,SPC(10);WO$(I,5);                  <195>
5160 IF WO$(I,6)<>""THEN PRINT#1,"; "WO$(I
     ,6);                                       <111>
5170 GOTO 5050                                  <222>
5180 PRINT#1                                    <079>
5190 PRINT#1,"{2SPACE}";WA$(WO(I));"{2SPAC
     E}";                                       <059>
5200 PRINT#1,WO$(I,2);                          <232>
5210 IF WO$(I,3)<>""THEN PRINT#1,"; ";WO$(
     I,3);                                      <216>
5220 PRINT#1:PRINT#1,SPC(10);:GOTO 5020         <234>
5230 PRINT LL$"{HOME,2DOWN,2RIGHT,GREY 3}A
     BBRUCH":PRINT"{UP,3RIGHT}< TASTE DRUE
     CKEN{SHIFT-SPACE}>"                        <249>
5240 TT=12:GOSUB 6250                           <250>
5250 GET A$:IF A$=""THEN 5250                   <221>
5260 POKE SI+4,0                                <232>
5270 GOTO 4670                                  <218>
5280 PRINT LL$"{HOME,2DOWN,2RIGHT}FERTIG."      <234>
5290 PRINT"{UP,3RIGHT}< TASTE DRUECKEN >"       <136>
5300 TT=45:GOSUB 6250                           <087>
5310 GET A$:IF A$=""THEN 5310                   <087>
5320 POKE SI+4,0                                <036>
5330 GOTO 4670                                  <022>
5340 :                                          <236>
5350 :                                          <246>
5360 REM **                                     <073>
5370 REM ** UNTERPROGRAMME                      <130>
5380 REM **                                     <095>
5390 :                                          <032>
5400 REM ** UMRECHNUNG ZAHL-USING(3)            <098>
5410 WX$=STR$(WX):LW=LEN(WX$):WX$=RIGHT$(W
     X$,LW-1):WX$=LEFT$("{2SPACE}",4-LW)+W
     X$                                         <233>
5420 RETURN                                     <144>
5430 :                                          <072>
5440 REM ** UMRECHNUNG ZAHL-USING(2)            <134>
5450 WX$=STR$(WX):LW=LEN(WX$):WX$=RIGHT$(W
     X$,LW-1):WX$=LEFT$("0",3-LW)+WX$           <040>
5460 RETURN                                     <184>
5470 :                                          <112>
5480 REM ** WAHL DER WORTART                    <238>
5490 PS=56187:AW=1                              <028>
5500 PA=PS+AW*4:POKE PA,1:POKE PA+1,1:POKE
      PA+2,1                                    <043>
5510 GET A$:IF A$=""THEN 5510                   <162>
5520 IF A$<>","AND A$<>"."AND A$<>CHR$(13)
     THEN 5510                                  <169>
5530 POKE PA,14:POKE PA+1,14:POKE PA+2,14       <246>
5540 IF A$=CHR$(13)THEN RETURN                  <003>
5550 IF A$=","THEN AW=AW-1:IF AW=0 THEN AW
     =6                                         <100>
5560 IF A$="."THEN AW=AW+1:IF AW=7 THEN AW
     =1                                         <136>
5570 GOTO 5500                                  <032>
5580 :                                          <222>
5590 REM ** EINGABE: INDEX & WOERTER            <089>
5600 ZZ=ES                                      <236>
5610 IF ZZ=0 THEN PRINT LEFT$(DD$,7)SPC(32
     )ZZ$(0);:GOTO 5630                         <116>
5620 PRINT LEFT$(DD$,8+2*ZZ)SPC(16)ZZ$(ZZ)
     ;                                          <206>
5630 PRINT"@";                                  <251>
5640 GET A$:IF A$=""THEN 5640                   <232>
5650 IF A$=CHR$(13)THEN 5760                    <105>
5660 IF A$=CHR$(20)THEN 5840                    <015>
5670 IF A$=CHR$(19)THEN 5800                    <041>
5680 IF A$=CHR$(141)THEN 5780                   <016>
5690 IF A$=CHR$(133)THEN 5890                   <225>
5700 IF A$=CHR$(134)THEN 5860                   <110>
5710 IF A$<" "OR A$>"Z"THEN 5640                <028>
5720 IF A$>CHR$(122)AND A$<CHR$(192)THEN 5
     640                                        <176>
5730 IF ZZ(ZZ)=20 THEN 5640                     <201>
5740 IF ZZ(ZZ)=5 AND ZZ=0 THEN 5640             <037>
5750 ZZ$(ZZ)=ZZ$(ZZ)+A$:ZZ(ZZ)=ZZ(ZZ)+1:PR
     INT"{LEFT}"A$"@";:GOTO 5640                <156>
5760 ZZ=ZZ+1:IF ZZ=7 THEN ZZ=0+ES               <007>
5770 PRINT"{LEFT,SPACE}":GOTO 5610              <130>
5780 ZZ=ZZ-1:IF ZZ=-1+ES THEN ZZ=6              <115>
5790 PRINT"{LEFT,SPACE}":GOTO 5610              <150>
5800 ZZ$(ZZ)="":ZZ(ZZ)=0                        <034>
5810 IF ZZ=0 THEN PRINT LEFT$(DD$,7)SPC(32
     )"@{5SPACE}";:GOTO 5610                    <008>
5820 PRINT LEFT$(DD$,8+2*ZZ)SPC(16)"@{20SP
     ACE}";:GOTO 5610                           <198>
5830 GOTO 5640                                  <182>
5840 IF ZZ(ZZ)=0 THEN 5610                      <113>
5850 ZZ$(ZZ)=LEFT$(ZZ$(ZZ),ZZ(ZZ)-1):ZZ(ZZ
     )=ZZ(ZZ)-1:PRINT"{LEFT,SPACE}":GOTO 5
     610                                        <035>
5860 IF ES=2 THEN ET=1                          <146>
5870 FOR I=1+ET TO 6:ZZ$(I)="":ZZ(I)=0:PRI
     NT LEFT$(DD$,8+2*I)SPC(16);                <072>
5880 PRINT"{20SPACE}":NEXT:ZZ=0+ES:ET=0:GO
     TO 5610                                    <098>
5890 IF ES=1 THEN RETURN                        <192>
5900 FOR I=0 TO 6:WO$(Q1,I)=ZZ$(I):NEXT:RE
     TURN                                       <193>
5910 :                                          <044>
5920 REM ** EINGABE FILENAME                    <115>
5930 FL$="":PRINT LEFT$(DD$,10)SPC(16)"{21
     SPACE}"                                    <128>
5940 PRINT LEFT$(DD$,10)SPC(16)FL$"@";          <094>
5950 GET A$:IF A$=""THEN 5950                   <097>
5960 IF A$=CHR$(19)THEN 5930                    <045>
5970 IF A$=CHR$(20)THEN 6040                    <086>
5980 IF A$=CHR$(13)THEN PRINT"{LEFT,SPACE}
     ":RETURN                                   <131>
5990 IF A$=CHR$(34)THEN 5950                    <196>
6000 IF A$<" "OR A$>"Z"THEN 5950                <074>
6010 IF A$>"←"AND A$<"A"THEN 5950               <215>
6020 IF LEN(FL$)=13+ES THEN 5950                <077>
6030 FL$=FL$+A$:GOTO 5940                       <118>
6040 FL=LEN(FL$):IF FL<2 THEN 5930              <221>
6050 FL$=LEFT$(FL$,FL-1):PRINT"{LEFT,SHIFT
     -SPACE}";:GOTO 5940                        <192>
6060 :                                          <194>
6070 REM ** EINGABE KONTROLLE                   <099>
6080 PRINT LL$"{HOME,2DOWN,2RIGHT}FILENAME
      EINGEBEN."                                <064>
6090 GOSUB 5920                                 <188>
6100 IF FL$=""THEN RETURN                       <018>
6110 PRINT LL$"{HOME,2DOWN,2RIGHT}INDEX EI
     NGEBEN."                                   <205>
6120 PRINT"{UP,2RIGHT,RVSON,LIG.BLUE,SPACE
     }+{SPACE,RVOFF,GREY 3,SPACE}10 EINH.
     VOR"TAB(60)"{RVSON,LIG.BLUE,SPACE}-{S
     PACE,RVOFF,GREY 3,SPACE}1 EINH. VOR"       <139>
6130 GET A$:IF A$=""THEN 6130                   <050>
6140 IF A$="+"OR A$="-"THEN ES=1:GOTO 2070
     :ES=0                                      <036>
6150 IF A$=CHR$(13)THEN PRINT LL$:RETURN        <033>
6160 GOTO 6130                                  <156>
6170 :                                          <050>
6180 REM ** INFO-MASKE                          <059>
6190 POKE BA,11                                 <075>
6200 PRINT"{WHITE,CLR}"T1$T3$T5$T5$T4$T3$;
     :FOR I=1 TO 17:PRINT T5$;:NEXT:PRINT
     T4$T2$;                                    <151>
6210 POKE 2023,125                              <245>
6220 RETURN                                     <182>
6230 :                                          <110>
6240 REM ** SOUND                               <126>
6250 POKE SI+24,15:POKE SI+5,11:POKE SI+6,
     0:POKE SI,0:POKE SI+1,TT:POKE SI+4,17      <114>
6260 RETURN                                     <222>
6270 POKE SI+24,15:POKE SI+5,28:POKE SI+6,
     0:POKE SI,0:POKE SI+1,2:POKE SI+4,33:
     RETURN                                     <166>
```


Listing »Cursus Latinus«. (Schluß)

Hypra-Text

Mit Hypra-Text haben Sie ein sehr leistungsfähiges Programm vorliegen. Besonders die Arbeit mit Textvariablen und die umfangreichen Möglichkeiten bei der Druckeransteuerung sind hervorzuheben. Hypra-Text gehört sicherlich zu den besten Programmen, die man abtippen kann.

Tippen Sie zuerst das Hauptprogramm (Listing 1) ab und speichern es auf Diskette. Anschließend ist noch eine Maschinenroutine »PROFIP V1.0« (Listing 2) und das Programm »Zloader« (Listing 3) einzugeben. Die Maschinenroutine übernimmt alle Aufgaben, die in Basic zu lange dauern würden. Das Programm »Zloader« ist ein in sich geschlossenes Basic-Programm, das es Ihnen erleichtert, einen beliebigen Drucker an das Programm anzupassen. So läßt sich mit diesem Programm zum Beispiel sehr leicht der Code eines Zeichens ändern, das zum Drucker gesendet wird. Um Ihnen Tipparbeit zu ersparen, ist in diesem Programm eine Routine integriert, die einen Großteil des Zloaders selbst generiert. Haben Sie das Programm »Zloader« abgetippt, ist es das erste Mal mit »RUN20000« zu starten. Auf dem Bildschirm erscheinen nun nacheinander die erzeugten Basic-Zeilen. Meldet sich das Programm mit READY, sind zwei Tabellen erzeugt worden.

Die erste Tabelle steht in den Zeilen 8000 bis 8500. Hier wird jedem Zeichen, das hinter der DATA-Anweisung steht, ein Drucker-Code zugeordnet. Sie werden feststellen, daß in den Zeilen 8000 bis 8060 statt des Zeichens nur ein Punkt steht. Dieser Punkt steht für die Tastenkombinationen, die zusammen mit der »CTRL-Taste« erreicht werden. So bedeutet der erste Punkt in Zeile 8000, hinter dem eine 1 steht, »CTRL A«. Der zweite in Zeile 8002, hinter dem eine 2 steht, bedeutet folglich »CTRL B«. Möchten Sie zum Beispiel einen Drucker an den User-Port über die im Programm »Profiprint« (erscheint in der nächsten Ausgabe) integrierte Centronics-Schnittstelle betreiben, so sind in den Zeilen 8000 bis 8500 alle Codes für Großbuchstaben durch die entsprechenden Codes für Kleinbuchstaben und die Codes für Kleinbuchstaben durch die entsprechenden Codes für Großbuchstaben zu ersetzen. Der Grund dafür ist, daß die meisten Drucker, die über eine Centronics-Schnittstelle betrieben werden, die Zeichen im ASCII-Format drucken. Der Commodore-Zeichensatz weicht jedoch insofern von dem Standard-ASCII-Zeichensatz ab, daß die Codes für Groß- und Kleinbuchstaben vertauscht sind. Achten Sie bei der Anpassung des Zeichensatzes darauf, daß im Zeilenbereich 8000 bis 8500 weder Zeilen hinzugefügt noch gelöscht werden dürfen. Auch ist es unzulässig, Zeichen zwischen den Anführungszeichen zu ändern.

Ein besonderer Leckerbissen befindet sich in der zweiten Tabelle, in den Zeilen 8500 bis 9000. Hier werden alle Umlaute und Sonderzeichen sowohl für den Bildschirm wie auch für den Drucker definiert.

Ersetzen Sie die erzeugten Zeilen durch das Listing 4. Denn in diesem Listing wurden die Steuerzeichen so definiert, wie sie in der Anleitung zu »Hypra-Text« erklärt werden. Sollten Ihnen die auf dem Bildschirm erzeugten Umlaute und Sonderzeichen nicht gefallen, oder stimmen die Drucker-Steuerzeichen nicht mit Ihrem Drucker überein, so besteht die Möglichkeit, die Umlaute, Sonderzeichen und Drucker-Steuerzeichen dem eigenen Bedarf anzupassen. Die Zeilen in Listing 4 sind ähnlich aufgebaut, wie die von 8000 bis 8500. Hier steht jedoch nicht nur ein Code, sondern acht Codes für das Bildschirm-Zeichen und acht Codes für das Drucker-Steuerzeichen. Die acht Codes für ein Bildschirmzeichen werden wie folgt erstellt:

1. Zeichnen Sie ein 8x8-Punkte-Raster (Bild 1).
2. Wo später ein Punkt auf dem Bildschirm gesetzt werden soll, füllen Sie das entsprechende Kästchen aus.
3. Addieren Sie die Spaltenwerte der ausgefüllten Kästchen und schreiben Sie sie hinter die Zeile.
4. Übernehmen Sie die Zeilenwerte der Reihe nach in die DATA-Zeile.

Beispiel:
8534 DATA "{Commodore-E}",102,0,60,102,102,102,60,0,185,196,196,196,185,128,0,0
definiert das Bildschirm- und Druckerzeichen »ö«.
Die Zahlenreihe »102,0,60,102,102,102,60,0,« definiert das Bildschirm-Zeichen,
die Zahlenreihe »185,196,196,196,185,128,0,0« definiert das Drucker-Zeichen.

Bestimmte Drucker-Steuerzeichen verlangen weniger als acht Codes. So bedeutet zum Beispiel 27, 45, 49 »unterstreichen ein« auf einem Epson-Drucker. Soll eine solche Befehlssequenz an den Drucker übergeben werden, so ist diese nach der Definition des Bildschirmzeichens unmittelbar einzugeben. Der Rest der Zeile ist mit Nullen aufzufüllen. In diesem Fall mit fünf Nullen.
Beispiel:
8500 DATA "(Zeichen)",1,2,3,4,5,6,7,8,27,45,49,0,0,0,0,0

Es müssen also nach der Definition des Bildschirmzeichens immer acht Codes für das Drucker-Steuerzeichen folgen, auch dann, wenn weniger Codes an den Drucker übergeben werden. Nullen im Drucker-Steuerzeichen sind durch den ASCII-Wert 48 zu ersetzen.

Wollen Sie das Bildschirm-Zeichen beibehalten, so ist nach dem in Anführungsstrichen stehenden Zeichen die Zahl 999, gefolgt von den acht Drucker-Codes, einzugeben. Alle Codes müssen, wie üblich bei DATA-Anweisungen, durch ein Komma getrennt werden.
Beispiel:
8500 DATA "(Zeichen)",999, 27,45,49,0,0,0,0,0

Wenn Sie in den Zeilen 8500 bis 9000 etwas ändern, ist darauf zu achten, daß Zeilen weder gelöscht, übersprungen oder hinzugefügt werden dürfen. Auch ist unbedingt darauf zu achten, daß die Reihenfolge der Steuerzeichen und Umlaute eingehalten wird. Denn das in der nächsten Ausgabe erscheinende Programm »Profiprint« erkennt aufgrund der Reihenfolge, ob es sich um ein Drucker-Sonderzeichen oder einen Umlaut handelt.

Haben Sie den Zeichensatz Ihren Wünschen angepaßt, ist das Programm »ZLOADER« nochmals mit »RUN« zu starten. Auf dem Bildschirm erscheinen jetzt nacheinander die neu definierten Sonderzeichen und Umlaute. Gefällt Ihnen eines der Zeichen nicht, so können Sie das Programm mit der RUN/STOP-Taste jederzeit unterbrechen, die entsprechende DATA-Zeile ändern und das Programm noch einmal mit RUN starten. Sind alle Zeichen zu Ihrer Zufriedenheit definiert, erzeugt das Programm zwei Files auf Diskette. Einen Bildschirm-Zeichensatz mit dem Namen »ZEICHEN DIN« und einen Drucker-Steuersatz mit dem Namen »ZEICHEN DDIN«. Außerdem wird der Zloader unter dem Namen »LOADER DIN 2.1« mit den abgeänderten DATA-Zeilen gespeichert.

Noch eines vorweg: Im Hauptprogramm von Hypra-Text werden Sie gefragt, ob Sie mit der DIN- oder normalen C 64-(CBM-)Tastatur arbeiten möchten.

Der Unterschied zwischen den beiden Tastaturen liegt darin, daß bei der DIN-Tastatur die Tasten so belegt sind, wie Sie es von einer Schreibmaschine gewohnt sind. Es stimmen also

die Zeichen die auf dem Bildschirm erscheinen nicht mit dem überein, was auf den Tasten steht. Wählen Sie dagegen die CBM-Tastatur, so erscheint genau das auf dem Bildschirm, was auf den Tasten steht. Trotzdem existieren auch hier die deutschen Umlaute, die über die Commodoretaste plus E,R,T,Z,U,I erreichbar sind. Um aber mit der CBM-Tastatur arbeiten zu können, erwartet Hypra-Text zwei Files auf der Diskette, die noch nicht erwähnt wurden. Bei den beiden Files handelt es sich um »ZEICHEN B« und »ZEICHEN D«. Diese beiden Files können aber sehr einfach generiert werden. Ersetzen Sie einfach im Programm »Zloader« in Zeile 40040 »ZEICHEN DIN« durch »ZEICHEN B« und in Zeile 40001 »ZEICHEN DDIN« durch »ZEICHEN D«. Anschließend ist das Programm Zloader nochmals mit RUN zu starten. Die erforderlichen Files werden nun auf der Diskette generiert.

Im weiteren Verlauf dieser Anleitung wird Ihnen auffallen, daß vor jeder Funktion zwei Zahlen stehen, die jeweils durch einen Punkt getrennt sind. So lesen Sie zum Beispiel »1.1.Eingabe des Textes«. Um diese Funktion zu erreichen, drücken Sie vom Hauptmenü aus einfach zweimal die »1«.

Das Hauptprogramm von »Hypra-Text«

Bevor Sie die Arbeit mit Hypra-Text beginnen, überzeugen Sie sich davon, daß sich folgende Files auf Ihrer Diskette befinden:

1. HYPRA-TEXT (in Basic, oder was zu empfehlen ist, in compilierter Form. Da im Programm SYS-Aufrufe mit zusätzlicher Parameterübergabe enthalten sind, eignet sich zum Compilieren nur der Austro-Compiler. Die Leser-Service-Diskette enthält neben dem Basic-Programm auch das compilierte File.)
2. PROFIP V1.0 (Maschinenroutinen)
3. ZEICHEN DIN (Bildschirm-Zeichen)
4. ZEICHEN DDIN (Drucker-Steuerzeichen)

und wenn die CBM-Tastatur gewünscht ist:

5. ZEICHEN B (Bildschirm-Zeichen)
6. ZEICHEN D (Drucker-Steuerzeichen)

Bevor Sie das Programm mit RUN starten, müssen Sie eine persönliche Kennung und die Laufwerksnummer Ihrer Floppy eingeben. Die Kennung dient dem Datenschutz: Außer Ihnen hat niemand Zugriff auf persönliche Daten und Texte. Dies geschieht mit folgenden Befehlen:

POKE 930,ASC("1.Buchstabe"):POKE 931,ASC("2.Buchstabe") :POKE 932,Laufwerksnummer

Als Kennung können Sie zum Beispiel die Initialen Ihres Namens eingeben. Die Laufwerksnummer ist im Normalfall »8«.

In einer der nächsten Ausgaben wird, passend zu Hypra-Text, eine Adreßverwaltung erscheinen, die diese POKEs durch die automatische Abfrage eines Geheimwortes überflüssig macht.

Nun werden Sie gefragt, ob Sie mit der DIN- oder der CBM-Tastatur arbeiten wollen. Ist die DIN-Tastatur gewünscht, drücken Sie einfach auf die RETURN-Taste. Ansonsten ist das Wort »din« durch »cbm« zu ersetzen. Es werden dann automatisch die erforderlichen Files nachgeladen.

Hypra-Text erwartet jetzt die Eingabe des Datums im Format TTMMJJ. Der 1. Januar 1986 wird zum Beispiel mit 010186 angegeben.

1.Text erstellen

Nach Wahl dieses Punktes kommen Sie in ein Untermenü.

1.1.Eingabe des Textes

Die Eingabezeile, also die Zeile, in der der Cursor steht, befindet sich immer in der Mitte des Bildschirms. Nach jeder abgeschlossenen Zeile wird der gesamte Text um eine Zeile nach oben geschoben. Um einen eingetippten Text zu editieren, können Sie mit RETURN den Text eine Zeile nach unten und mit F5 eine Zeile nach oben schieben. Die Tasten »Cursor-rechts« und »Cursor-links« behalten ihre alte Funktion bei. Mit »Cursor-hoch« und »Cursor-runter« können Sie ein Wort vor- oder zurückspringen. Die HOME-Taste hat bei Hypra-Text die Funktion »Tabulator«. Dazu befindet sich in der fünften Bildschirmzeile von oben eine Tabulatorzeile. Wenn die Taste HOME gedrückt wird, so springt der Cursor bis zur nächsten »+«-Marke. Der Leerraum bei einem Tabulator-Sprung wird mit Punkten gefüllt, die aber nicht gedruckt werden.

Weiterhin sind die folgenden Tasten mit Drucker-Steuerzeichen belegt, vorausgesetzt, Listing 4 wurde wie abgedruckt in den Zloader übernommen:

Commodore-Taste und
n: Breitschrift an
m: Breitschrift aus
g: Fettschrift an
h: Fettschrift aus
j: Italic-Schrift an
k: Italic-Schrift aus
c: Unterstreichen an
v: Unterstreichen aus

Diese Steuerzeichen sind natürlich nur dann sinnvoll, wenn der angeschlossene Drucker sie auch ausführen kann.

Die CLR-Taste löscht die Zeile, in der sich der Cursor befindet.
SHIFT-RETURN markiert einen Absatz.
Die Funktionstasten sind wie folgt belegt:
F1: zum Hauptmenü (der erstellte Text bleibt natürlich erhalten)
F2: Einfügen. Man hat jetzt zwei Möglichkeiten, den Text einzufügen:
a) F3: Der Computer fügt ab der Cursor-Zeile die eingegebene Anzahl an Leerzeilen in den Text ein,
b) F5: Alles was rechts vom Cursor steht wird verschoben, und zwar so lang, bis eine Funktionstaste gedrückt wird.
F3: mehrere Zeilen löschen. Ab der Cursorzeile wird die eingegebene Anzahl an Zeilen gelöscht.
F4: Block kopieren. Die aktuelle Cursorposition wird nun als Blockanfang interpretiert; das Blockende wird mit den Tasten F5 und F7 verschoben und mit F3 markiert. Danach müssen Sie mit den selben Tasten angeben, an welche Stelle im Text dieser Block kopiert werden soll.
F5: Cursor eine Zeile hoch.
F6: Nach Drücken dieser Taste können Sie mit F5 oder F6 eine Textseite vor- oder zurückblättern.
F7: Diese Taste schaltet zwischen den Hilfstafeln im unteren Teil des Bildschirms hin und her.
F8: Text nachladen. Nach der Eingabe des Namens wird der Text geladen und ab der Cursor-Zeile eingefügt.

Die Druckerbefehle von Hypra-Text

Diese Befehle bewirken nur etwas, wenn der Text auf dem Drucker ausgegeben wird. Auf dem Bildschirm werden sie nicht ausgeführt! Alle Befehle werden direkt in den Text hineingeschrieben.

noformat
Druckt den Text so, wie er auf dem Bildschirm steht.
mittezen
Druckt den Text mittezentriert.
rechtsfo
Druckt den Text rechtsbündig.
formnorm
Hebt »mittezen« und »rechtsfo« auf.
dezlen=xy
Dient zur Formatierung von Zahlen innerhalb des Textes.
dezlen=34 bedeutet zum Beispiel, daß alle Zahlen auf drei

Stellen (=x) vor, und vier Stellen (=y) nach dem Komma formatiert werden.

1.2.Eingabe des Textnamen
Der hier eingegebene Name kann in Punkt 5.5. durch »=« übernommen werden. Er wird im Hauptmenü angezeigt.

1.3.Eingabe über Maske
Die Erstellung einer Maske (zum Beispiel für Serienbriefe) ist äußerst einfach.

Sie tippen im Texteingabemodus Ihre Maske als normalen Text ein. An alle Stellen, an denen Sie später zum Beispiel eine Adresse eingeben wollen, setzen Sie eine Markierung (Commodore-Taste und »Klammeraffe«). Dann speichern Sie die fertige Maske ab, und wählen Punkt 1.3.

1.4.Text löschen
Es wird der gesamte, im Speicher befindliche Text gelöscht. Der Computer zeigt dabei an, welche Zeile er gerade löscht. Sicherheitshalber verlangt er vorher noch eine Bestätigung.

1.5.Text ausfüllen
Hier können Sie an alle Zeilen Punkte oder bestimmte Zeichen anhängen. Ob links oder rechts vom Text, und ab welcher Zeile bestimmen Sie. Dies ist zum Beispiel sinnvoll, wenn Sie eine Tabelle haben und an jede dritte Zeile einen Namen oder ähnliches anhängen wollen.

Wenn Sie die Frage »Neben den Text (j/n)« mit »n« beantworten, so werden alle angegebenen Zeilen gelöscht und durch die Änderung ersetzt.

1.6.Eingabe Tabulatorzeile
Hier können Sie die Tabulatorzeile des Texteingabemodus ändern. Dazu benützen Sie die Cursor-Tasten. Mit »+« markieren Sie einen Tabulator-Stop.

2.1.Text ändern
Entspricht der Eingabefunktion (Punkt 1.1.). Sie geben nur zusätzlich mit an, ab welcher Zeile Sie ändern wollen.

2.2.Worttausch
Sie können den Text nach einem bestimmten Wort durchsuchen und durch ein anderes ersetzen lassen. Der Computer fragt Sie nach jedem gefundenen Wort, ob Sie es ändern wollen.

2.3.Text löschen
Identisch mit Punkt 1.4.

3.Profiprint laden
Um einen Text auszudrucken, benötigt man das Programm Profiprint. Dieses wird in der nächsten Ausgabe veröffentlicht.

4.Bildschirmausgabe
Über diesen Punkt können Sie sich eine Übersicht über Ihren Text verschaffen. Es wird neben dem Text eine Zeilenmarkierung ausgegeben. (Diese Funktion ist auch mit F3 erreichbar.)

5.1.Text vergleichen
Der im Arbeitsspeicher befindliche Text wird mit dem gleichnamigen auf Diskette verglichen.

Diese Funktion ist auch hervorragend geeignet, um sich einen auf Diskette befindlichen Text anzusehen, ohne den gerade bearbeiteten zu löschen. Der Computer druckt nämlich alle Zeilen des Textes aus, die nicht mit dem Text im Speicher übereinstimmen. Nach jeder Zeile ist eine Taste zu drücken.
F1 bricht diesen Vorgang vorzeitig ab.

5.2.Text umbenennen
Entspricht dem RENAME-Befehl der Floppy. Man kann einem auf Diskette befindlichen Text einen anderen Namen geben.

5.3.Text löschen
Entspricht dem SCRATCH-Befehl der Floppy. Ein Text wird von Diskette gelöscht.

5.4.Text laden
Ein Text wird von Diskette geladen. Sie haben dabei die Wahl zwischen:

5.4.1.Text löschen und neuen Text laden
Der Arbeitsspeicher wird zuerst gelöscht, und dann der gewünschte Text geladen.

5.4.2.Text hinter alten Text laden
Der zu ladende Text wird an den Text angehängt, der im Arbeitsspeicher steht.

5.5.Text speichern
Der Inhalt des Arbeitsspeichers wird auf Diskette abgelegt.

5.6.Inhaltsverzeichnis
Es wird dabei nicht das gesamte Directory der Diskette ausgegeben, sondern nur alle Texte mit Namen, Datum der Eingabe, und der persönlichen Kennung, die verwendet wurde. (POKE 930,...)
Dabei ist wichtig, daß man nur auf seine eigenen Texte zugreifen kann.

6.Rechnen
Hier ist ein kleiner Taschenrechner eingebaut. Die Rechenergebnisse werden in drei Variablen gespeichert, und können im Text verwendet werden.

Textvariable

7.Variablen
Die Arbeit mit Textvariablen ist eine der ganz großen Stärken von Hypra-Text. Wenn Sie einen Text schreiben, in dem zum Beispiel sehr oft das Wort »Computer« vorkommt, dann speichern Sie dieses Wort einfach in einer der 50 (fünfzig!) Textvariablen und wählen bei der Texterstellung diese Variable durch » @ « und einer zweistelligen Variablennummer an. (Aus Variable 1 wird 01 usw.) Hypra-Text ersetzt diese Variablennummer nach dem nächsten RETURN automatisch durch ihren Inhalt. Voraussetzung ist allerdings, daß Sie mit Punkt 7.3. diese Funktion aktiviert haben.

7.1.Variablen zeigen
Der Inhalt aller fünfzig Variablen wird auf dem Bildschirm ausgegeben. Durch beliebigen Tastendruck können Sie weiterblättern.

7.2.Variablen eingeben
Der Inhalt einer Variable darf maximal 40 Zeichen lang sein.

7.3.Variablen an
Damit wird Hypra-Text mitgeteilt, daß man mit Textvariablen arbeiten möchte.

7.4.Variablen aus
Die Variablenfunktion wird ausgeschaltet.

7.5.Variablen in Text umwandeln
Man hat zum Beispiel einen Serienbrief erstellt. Für die Adresse wurden Variablen verwendet. Diese sollen erst später durch einen entsprechenden Inhalt ersetzt werden.

Man läßt also bei der Texterstellung die Variablen ausgeschaltet (7.4.), und speichert den Text. Später gibt man unter Punkt 7.2. zum Beispiel eine bestimmte Adresse in die Variablen ein. Dann wählt man Punkt 7.5. Hypra-Text ersetzt jetzt die Variablen durch Klartext.

7.6.Variablen in Textspeicher schreiben
Wenn Sie eine Anzahl von Variablen erstellt haben, so möchten Sie diese natürlich zur späteren Verwendung speichern. Dieser Menüpunkt schreibt den Inhalt der Variablen in den Textspeicher in die Zeilen 1-50. Von dort aus können Sie die Variablen dann wie einen normalen Text speichern.

7.7.Textspeicher in Variablen schreiben
Um einmal definierte Variablen später wieder verwenden zu können, laden Sie den unter 7.6. gespeicherten Variablentext in den Arbeitsspeicher und rufen dann diesen Punkt auf. (Variablen anschalten nicht vergessen!)

7.8.Variablen aus Dateiverwaltung
In einer der nächsten Ausgaben wird eine Dateiverwaltung veröffentlicht. Mit diesem Menüpunkt können Sie Daten aus der Dateiverwaltung in Hypra-Text übernehmen. Genauere Angaben folgen.

(Oliver Fendt/ah/tr)



	128	64	32	16	8	4	2	1	SUMME
ZEILE 1		■	■			■	■		102
ZEILE 2									0
ZEILE 3			■	■	■	■			60
ZEILE 4		■	■			■	■		102
ZEILE 5		■	■			■	■		102
ZEILE 6		■	■			■	■		102
ZEILE 7			■	■	■	■			60
ZEILE 8									0

Bild 1. DATA "{Commodore-E}", 102,0,60,102,102,102, 60,0 definiert das Bildschirmzeichen »ö«.

```
0 GOTO 10                                 <178>
1 IF PEEK(198)>0 THEN GET A$:RETURN       <214>
2 GOSUB 30600:GOTO 1                      <207>
3 PRINT"{4DOWN,SPACE}+ BITTE WARTEN +
  ICH{2SPACE}A R B E I T E{2SPACE}+
  ":RETURN                                <158>
4 POKE 908,0:POKE 918,30:POKE 920,11:
  POKE 917,36:POKE 924,0:A$="":RETURN <088>
5 SYS 50176,A$,A$:F=PEEK(913):RETURN  <047>
9 GOTO 30001                              <117>
10 GOTO 22000                             <014>
11 GOSUB 31000:POKE 650,128:POKE 2,18
   4                                      <153>
12 GOSUB 40000                            <016>
14 PRINT:L$=A6$                           <087>
15 KL=1:PRINT CHR$(14)                    <087>
17 DIM A$(600):AN=0:MAX=600               <050>
26 PRINT"{CLR}";PT$;"{DOWN}"              <242>
27 GOTO 100                               <211>
30 A$=LEFT$(A$,LEN(A$)-1):RETURN          <148>
31 IF RIGHT$(A$,1)=" "THEN GOSUB 30:G
   OTO 31                                 <074>
32 RETURN                                 <029>
40 A$=LEFT$(A$(T),LEN(A$(T))-1):RETUR
   N                                      <122>
70 REM ** UNTERPROGRAMM **                <138>
71 PRINT:PRINT"GEBEN SIE EIN,VON WELC
   HER ZEILENNR{6SPACE}(0=ANFANG)"        <152>
72 INPUT V:IF V=0 THEN V=1                <146>
73 PRINT"GEBEN SIE EIN,BIS ZU WELCHER
   ZEILENNR{3SPACE}(0=ENDE)"              <042>
74 INPUT B:IF B>MAX THEN PRINT"ZU GRO
   SS !":GOTO 74                          <031>
75 IF B=0 THEN B=AN                       <051>
79 RETURN                                 <137>
80 PRINT"{HOME,19DOWN}":RETURN            <178>
90 PRINT LEFT$(A6$,MB):RETURN             <219>
95 GOSUB 80:FOR I=1 TO 4:GOSUB 90:NEX
   T:GOSUB 80:RETURN                      <150>
100 PRINT"{CLR}";PT$;:CLOSE 2:CLOSE 1
    5:CLOSE 5:POKE 198,0:BA=0:DR=0:BR
    =0:SP=0                               <084>
101 IF AN/MAX>0.7 THEN B1$="{RVSON}"+
    STR$(INT((AN/MAX)*100))+" % VOLL
    !!!{RVOFF}":X=FRE(9)                  <074>
102 IF AN/MAX>0.7 THEN B2$="{RVSON}[5
    ] = SPEICHERN{RVOFF}":B3$="{RVSON
    }ACHTUNG{RVOFF}":SP=1                 <141>
103 V$(50)=D$:V$(47)=BA$:V$(46)=STR$(
    RR):V$(45)=STR$(RR+RR*(MW/100))       <244>
104 V$(44)=STR$(RR*(MW/100))              <203>
105 POKE 54296,0:IF PEEK(1022)=200 TH
    EN AM=2                               <212>
107 PRINT"{HOME}";PT$;                    <213>
109 POKE 53265,27                         <128>
110 PRINT"{DOWN,2SPACE}DATUM: ";D$        <052>
111 GOSUB 300:PRINT"{HOME,3DOWN}"         <010>
112 PRINT" ={SPACE,RVSON}SYSTEM{4SPAC
    E,RVOFF,SPACE}:";B1$                  <233>
113 PRINT" ={SPACE,RVSON}EMPFEHLUNG{R
    VOFF,SPACE}:";B2$                     <207>
114 PRINT" ={SPACE,RVSON,10SPACE,RVOF
    F,SPACE}:";B3$                        <069>
115 PRINT"{DOWN,SPACE}= TEXT NAME{2SP
    ACE}:";T1$                            <179>
116 PRINT" = PLATZ{6SPACE}:";MAX-AN;"
    ZEILEN FREI"                          <144>
120 REM ** MENU **                        <027>
125 PRINT"{2DOWN,3RIGHT}-1-{2SPACE}TE
    XT ERSTELLEN"                         <209>
130 PRINT"{3RIGHT}-2-{2SPACE}TEXT AEN
    DERN"                                 <249>
132 PRINT"{3RIGHT}-3-{2SPACE}PROFIPRI
    NT LADEN"                             <212>
134 PRINT"{3RIGHT}-4-{2SPACE}BILDSCHI
    RMAUSGABE"                            <159>
135 PRINT"{3RIGHT}-5-{2SPACE}ARBEITEN
     MIT DISKETTE "                       <087>
136 PRINT"{3RIGHT}-6-{2SPACE}";FU$        <088>
137 PRINT"{3RIGHT}-7-{2SPACE}ARBEITEN
     MIT VARIABLEN"                       <150>
145 PRINT"{2DOWN,2RIGHT}DIE FUNKTIONS
    TASTEN"                               <089>
147 PRINT"{2RIGHT}F1= EINGABE F3=BILD
    SCHIRMAUSGABE"                        <003>
148 PRINT"{2RIGHT}F5= ENDE{3SPACE}";      <051>
149 IF SP=1 THEN GOSUB 30601              <100>
150 GET A$:IF A$=""THEN 105               <153>
154 IF A$="{F3}"THEN 7000                 <012>
155 IF A$="{F1}"THEN 1015                 <036>
156 IF A$="{F5}"THEN 8000                 <146>
197 IF VAL(A$)<1 OR VAL(A$)>8 THEN 15
    0                                     <085>
199 X=VAL(A$):ON X GOTO 1000,2000,561
    00,7100,5000,20000,9700               <068>
200 GOTO 100                              <128>
260 RETURN                                <064>
300 REM ** BS AUFBAU **                   <051>
301 IF BA=1 THEN RETURN                   <134>
310 A1$=" R***************************
    **********5"                          <017>
320 A2$=" G***************************
    **********W"                          <200>
330 A3$=" 7***************************
    **********X"                          <043>
340 A4$=" ={35RIGHT}="                    <213>
350 PRINT"{HOME,2DOWN}":PRINT A1$:PRI
    NT A4$:PRINT A4$:PRINT A4$:PRINT
    A2$                                   <104>
360 PRINT A4$:PRINT A4$:PRINT A2$         <100>
370 FOR T=1 TO 9:PRINT A4$:NEXT:PRINT
     A2$                                  <143>
380 PRINT A4$:PRINT A4$:PRINT A4$:PRI
    NT A3$;                               <173>
499 BA=1:RETURN                           <046>
1000 PRINT"{CLR}";PT$                     <116>
1001 PRINT"{HOME,2DOWN,RVSON,SPACE}*
     * TEXTPROGRAMM{5SPACE}* *{2SPACE
     }EINGABE{2SPACE}* *{SPACE,DOWN,R
     VOFF,2DOWN}";:TA=0:TA$=""            <177>
1003 PRINT"{RVSON}=1={2SPACE}EINGABE
     DES TEXTES"                          <070>
1004 PRINT"=2= EINGABE DES TEXTNAMES" <136>
1005 PRINT"=3= EINGABE UEBER MASKE"   <058>
1007 PRINT"=4= TEXT LOESCHEN"             <083>
1008 PRINT"=5= DEN TEXT AUSFUELLEN":P
     RINT"=6= EINGABE TABULATORZEILE" <015>
1009 GOSUB 1                              <182>
1010 IF A$="0"OR A$="{F1}"THEN 100        <060>
1011 B1$="AENDERN [2]":B2$="SPEICHERN
      [5]":T=VAL(A$)                      <114>
1013 IF T=0 THEN 1009                     <241>
1014 ON T GOTO 1015,1850,33000,1300,9
     400,1830,100                         <146>
1015 PRINT"{CLR}";PT$:PRINT"{HOME,2DO
     WN,RVSON,SPACE}* *{2SPACE}TEXTPR
     OGRAMM{4SPACE}* *{2SPACE}EINGABE
     {2SPACE}* *{SPACE,DOWN,RVOFF}";:
     TA=0                                 <178>
1016 B1$="AENDERN{3SPACE}[2]"             <042>
1017 B2$="SPEICHERN [5]"                  <237>
1018 B3$="ANSEHEN{3SPACE}[F3]"            <120>
1020 A=AN:GOTO 2014                       <194>
1040 IF K1$="{RVSON}TASTEN{RVOFF,SPAC
     E}F7=HELP{11SPACE}F5=ZEILE HOCH"
     THEN 1050                            <226>
1046 K1$="{RVSON}TASTEN{RVOFF,SPACE}F
     7=HELP{11SPACE}F5=ZEILE HOCH"        <041>
1047 K2$="HOME = TABULATOR{4SPACE}RET
     URN=ZEILE RUNTER"                    <151>
1048 K3$="CRSR HOCH = WORT VOR{4SPACE
     }:CRSR => RECHTS"                    <011>
1049 K4$="CRSR RUNTER=WORT ZURUECK:CR
     SR <= LINKS":RETURN                  <243>
1050 K1$="{RVSON}FUNKTIONST.{RVOFF,SP
     ACE}F1=ENDE : F2=EINFUEGEN"          <131>
1051 K2$="F3= ZEILEN LOESCHEN : F4=BL
     OCK COPIEREN"                        <005>
1052 K3$="F5=ZEILE HOCH{7SPACE}: F6=S
     EITE +/-"                            <113>
1053 K4$="{RVSON}F7=HELP{3SPACE,RVOFF
     ,10SPACE}: F8=BLOCK LADEN":RETUR
     N                                    <142>
1100 FOR I=G8 TO MAX                      <071>
1110 GOSUB 1400                           <114>
1120 IF LEN(A$(I+1))<M8 THEN RETURN       <097>
1130 NEXT:RETURN                          <053>
1300 GOSUB 1301:GOTO 1000                 <174>
1301 PRINT"{CLR}";PT$;"{DOWN,RVSON,10
     SPACE}TEXT LOESCHEN{17SPACE}"        <146>
1302 PRINT"{DOWN}BIST DU SICHER? [J/(
     N)] ":GOSUB 1:IF A$<>"J"THEN 100 <079>
1305 PRINT"{2DOWN}":FOR T=1 TO MAX:PR
     INT"{UP}";T:A$(T)="":NEXT            <036>
1310 AN=0:RETURN                          <165>
1400 B=LEN(A$(I)):FOR A=M8 TO 1 STEP-
     1                                    <025>
1410 IF MID$(A$(I),A,1)=" "OR MID$(A$
     (I),A,1)="{SHIFT-SPACE}"THEN 143
     0                                    <129>
1420 NEXT:PRINT"ZEILE ZU LANG:FEHLER"
     :RETURN                              <024>
1430 A$(I+1)=RIGHT$(A$(I),B-A)+" "+A$
     (I+1):A$(I)=LEFT$(A$(I),A)           <065>
1450 RETURN                               <238>
1460 PRINT"{CLR}":FOR T=0 TO AN:PRINT
      A$(T):NEXT:PRINT"{2UP}"             <007>
1470 RETURN                               <002>
1830 PRINT"{CLR}";PT$                     <186>
1831 INPUT"MAXIMALE ZEILELAENGE (39)"
     ;M8:GOSUB 1832:GOTO 1000             <075>
1832 M8=M8+2:IF M8<10 THEN M8=39          <014>
1833 PRINT"{CLR}";PT$:PRINT"EINGABE T
     ABULATORZEILE":PRINT"!=START +=T
     AB.STOP -=NICHT                      <238>
1834 T7$=LEFT$(T7$,LEN(T7$)-1)            <003>
1835 A$=LEFT$(T7$,M8):GOSUB 9:T7$=A$  <029>
1840 M8=M8-2:T7$=LEFT$(T7$+"----------
     ----------------------------------"
     ,M8-1)                               <191>
1841 FOR T=1 TO LEN(T7$):IF MID$(T7$,
     T,1)="!"THEN CB=T                    <107>
1842 NEXT:T7$=T7$+"!":POKE 917,M8         <057>
1845 RETURN                               <125>
1850 PRINT"{CLR,4DOWN}TEXTNAME":A$=T1
     $:GOSUB 9:T1$=A$:GOTO 1000           <004>
2000 PRINT"{CLR}";PT$                     <100>
2001 PRINT"{HOME,2DOWN,RVSON,SPACE}+
     + TEXTPROGRAMM{5SPACE}+ +{2SPACE
     }AENDERN{2SPACE}+ +{SPACE,DOWN,R
     VOFF}";:PRINT"{DOWN,RVSON}AENDER
     N [1]{RVOFF}"                        <060>
2002 PRINT"{2DOWN,2SPACE}>0< MENU":PR
     INT"{DOWN,2SPACE}>1< TEXT AENDER
     N"                                   <071>
2003 PRINT"{DOWN,2SPACE}>2< WORT TAUS
     CH"                                  <000>
2005 PRINT"{DOWN,2SPACE}>3< TEXT LOES
     CHEN{2DOWN}"                         <057>
2006 GOSUB 1                              <163>
2008 IF A$="0"OR A$="{F1}"THEN 100        <042>
2009 B1$="SPEICHERN [5]":B2$="ANSEHEN
     {3SPACE}[F3]":B3$="HILFE{5SPACE}
     [F7]"                                <251>
2010 T=VAL(A$):IF T<1 OR T>3 THEN 200
     6                                    <145>
2011 ON T GOTO 2012,2900,1300             <109>
2012 GOSUB 80:PRINT" 0 = ENDE : 1 = A
     NFANG"                               <209>
2013 INPUT"{DOWN}AB ZEILE";A:A=A-1:IF
      A=-1 THEN A=AN                      <079>
2014 GOSUB 1046                           <193>
2020 A=A+1:IF A>AN THEN AN=A              <190>
2030 IF A<1 THEN A=1                      <050>
2031 Y=T:A$=A$(A)                         <035>
2040 EE=1:A$=A$(A):G8=A:GOSUB 30000       <151>
2041 IF U9=1 THEN A$(A+1)=LEFT$(A6$,C
     B-1)+R$+" "+A$(A+1):U9=2             <050>
2050 A$(A)=A$:IF LEN(A$(A))>M8 THEN 2
     040                                  <020>
2060 IF A$(A)=">"THEN A$(A)=""            <205>
2061 PP=A:IF F=3 THEN 2600                <071>
2062 IF F=7 THEN GOSUB 1040:GOTO 2030 <043>
2063 IF IN=1 AND F<>0 THEN IN=0:GOSUB
      2560:IF F=1 THEN F=0                <148>
2064 IF F=1 THEN EE=0:GOTO 100            <121>
2065 IF F=6 THEN 2200                     <198>
2070 IF IN=1 THEN GOSUB 2350              <096>
2071 IF F=2 THEN 2500                     <079>
2072 IF F=4 THEN 2800                     <147>
2073 IF F=8 THEN 2100                     <014>
2075 IF F=5 THEN A=A-1:GOTO 2030          <091>
2080 A$=A$(A):Y=T:GOTO 2020               <227>
2100 PRINT"{HOME,2DOWN,RVSON,SPACE}*
     * TEXTPROGRAMM{4SPACE}* * BLOCK
     LADEN * *"                           <074>
2110 GOSUB 95:GOSUB 4:POKE 920,22         <111>
2120 PRINT"NAME DES BLOCKES":GOSUB 5:
     IF F=1 THEN 2020                     <185>
2121 A$=">"+BA$+"??????"+A$               <201>
2125 CLOSE 2:CLOSE 5:OPEN 2,FA,15:OPE
     N 5,FA,2,A$+",S,R":GOSUB 34000:I
     F EN<>0 THEN 2100                    <107>
2130 INPUT#5,Z:GOSUB 2530                 <103>
2140 FOR T=1 TO Z:GOSUB 34100:GOSUB 3
     4000:IF A$=">"THEN A$=""             <149>
2150 A$(PP+T)=A$:NEXT:CLOSE 2:CLOSE 5
     :A=A+Z:GOSUB 2560:GOTO 2020          <139>
2200 GOSUB 95                             <250>
2201 PRINT"F5= SEITE + "                  <170>
2202 PRINT"F6= SEITE -{4SPACE}"           <235>
2203 PRINT"{DOWN}BITTE WAEHLEN"           <240>
2210 GOSUB 1:IF A$="{F1}"THEN A=A-1:G
     OTO 2020                             <000>
2220 IF A$="{F5}"THEN A=A+10:GOTO 202
     0                                    <110>
2230 IF A$="{F6}"THEN A=A-12:GOTO 202
     0                                    <155>
2240 GOTO 2210                            <220>
2300 Z=1:GOSUB 2530                       <059>
2320 A$(A+1)=RIGHT$(A$(A),LEN(A$(A))-
     PEEK(925)):CC=999:IN=1               <019>
2330 A$(A)=LEFT$(A$(A),PEEK(925))         <048>
2340 IF LEFT$(A$(A+1),1)=" "THEN A$(A
     +1)=RIGHT$(A$(A+1),LEN(A$(A+1))-
     1):GOTO 2340                         <012>
2345 A$(A)=A$(A)+"{SHIFT-SPACE}":A=A-
     1:GOTO 2360                          <047>
2350 Z=1:GOSUB 2530                       <111>
2360 PRINT"{HOME,2DOWN,RVSON}!!! TEXT
     PROGRAMM !!! AUFSCHIEBEMODUS !!!
     ":RETURN                             <075>
2500 PRINT"{HOME,2DOWN,RVSON,SPACE}*
     * TEXTPROGRAMM{4SPACE}* * AUFSCH
     IEBEN * *":GOSUB 2510:GOTO 2020  <074>
2510 GOSUB 95                             <050>
2511 PRINT"F3 = X - ZEILEN{2SPACE}F5
     = BIS [F1]"                          <172>
2512 GOSUB 1:IF A$="{F1}"THEN 2560        <140>
2513 IF A$="{F5}"THEN 2300                <217>
2520 PRINT"WIEVIELE ZEILEN":INPUT Z:P
     P=PP-1:A=A-1                         <002>
2530 FOR O=AN+Z TO PP+Z STEP-1            <122>
2540 A$(O)=A$(O-Z):NEXT                   <195>
2550 FOR O=PP+1 TO PP+Z:A$(O)="":NEXT
     :AN=AN+Z                             <150>
2560 PRINT"{HOME,2DOWN,RVSON,SPACE}+
     TEXTPROGRAMM + EINGABE / AENDERN
      +{2SPACE}":RETURN                   <063>
2600 PRINT"{HOME,2DOWN,RVSON,SPACE}*
     * TEXTPROGRAMM{4SPACE}* * LOESCH
     EN{3SPACE}* * "                      <186>
2610 GOSUB 95                             <152>
2620 PRINT"WIEVIELE ZEILEN":INPUT Z   <193>
2630 FOR O=PP TO AN+Z                     <206>
2640 A$(O)=A$(O+Z):NEXT                   <025>
2650 FOR O=AN-Z TO AN:A$(O)="":NEXT:A
     N=AN-Z:GOSUB 2560:A=A-1:GOTO 202
     0                                    <211>
2800 PRINT"{HOME,2DOWN,RVSON,SPACE}*
     * TEXTPROGRAMM{4SPACE}* * COPIER
     EN{4SPACE}* *":Z=0:Z1=0              <178>
2810 GOSUB 95:PRINT"MARKIEREN SIE DAS
      UNTERE ENDE MIT [F3]"               <204>
2820 PRINT"[F1] = ENDE{3SPACE}:{2SPAC
```

```
      E)[F3] = MARKIEREN"                    <116>
2825 PRINT"[F5] = HOCH(3SPACE):(2SPAC
      E)[F7] = RUNTER"                       <216>
2829 Z=1                                     <052>
2830 IF Z+PP<1 OR Z+PP>MAX THEN Z=1          <232>
2831 G9=PP+Z-6:GOSUB 30705:POKE 781,1
      1:POKE 782,0:SYS 58636                 <076>
2832 PRINT"(RVSON)";A$(PP+Z);"(RVOFF)
      "                                      <082>
2833 GOSUB 1:IF A$="(F1)"THEN 2030           <014>
2834 IF A$="(F5)"THEN Z=Z-1                  <024>
2835 IF A$="(F7)"THEN Z=Z+1                  <122>
2836 IF A$="(F3)"THEN 2850                   <024>
2840 GOTO 2830                               <218>
2850 IF Z1=0 THEN Z1=Z:GOSUB 80:PRINT
      "MARKIEREN SIE WOHIN DER BLOCK S
      OLL(4SPACE)":GOTO 2830                 <148>
2860 Z2=PP+Z-1                               <079>
2881 G9=PP+Z-6:GOSUB 30705:POKE 781,1
      1:POKE 782,0:SYS 58636                 <126>
2886 FOR T=PP TO PP+Z1:A$(((T+1)-PP)+
      Z2)=A$(T):NEXT                         <043>
2887 IF AN<((T+1)-PP)+Z2 THEN AN=((T+
      1)-PP)+Z2                              <237>
2890 GOTO 7000                               <084>
2900 PRINT"(CLR,3DOWN)NACH WELCHEM WO
      RT SOLL GESUCHT WERDEN":A$=""          <142>
2901 GOSUB 9:C$=A$:IF F<>0 THEN 100          <076>
2902 PRINT"(CLR,3DOWN)WELCHES WORT SO
      LL DAFUER EINGESETZT(5SPACE)WERD
      EN":GOSUB 9:B$=A$                      <046>
2903 PRINT"(CLR)";PT$;"(DOWN,5SPACE)(
      J) = AENDERN(2SPACE)(N)=LASSEN(3
      DOWN)"                                 <222>
2910 L1=LEN(C$):L2=LEN(B$):L3=L2-L1          <247>
2920 FOR T=1 TO AN+1:X=0:G=1:CC=0:GOS
      UB 30400                               <020>
2921 IF CC=999 THEN GOSUB 2960:CC=0          <032>
2925 X=LEN(A$(T)):IF X=0 THEN NEXT:GO
      TO 100                                 <152>
2930 FOR G=1 TO X                            <110>
2935 IF MID$(A$(T),G,L1)<>C$THEN NEXT
      :NEXT:GOTO 100                         <005>
2939 PRINT"(HOME,6DOWN)";A6$;A6$;A6$         <089>
2940 PRINT"(HOME,6DOWN)";RIGHT$("(3SP
      ACE)"+STR$(T),3);" ";LEFT$(A$(T)
      ,G-1)+"(RVSON)"+C$+"(RVOFF)";          <099>
2941 PRINT RIGHT$(A$(T),LEN(A$(T))-G-
      LEN(C$)+1)                             <153>
2942 GOSUB 1:IF A$="N"THEN 2950              <225>
2945 A$=LEFT$(A$(T),G-1)+B$+RIGHT$(A$
      (T),X-(G+L1)+1):X=X+L3:GOSUB 304
      00:A$(T)=A$                            <151>
2946 IF CC=999 THEN GOSUB 2960
                                             <008>
2950 NEXT:NEXT:GOTO 100                      <112>
2960 A$(T)=A$:A$(T+1)=R$+" "+A$(T+1):
      IF LEN(A$(T+1))<MB THEN RETURN         <230>
2970 Z=1:PP=T+1:GOSUB 2530:A$=A$(T+1)
      :GOSUB 30400:A$(T+1)=A$:A$(T+2)=
      R$:RETURN                              <230>
4000 PRINT"(CLR)";PT$                        <068>
4001 PRINT"(RVSON,SPACE)+ VARIABLEN A
      US DER DATEIVERWALTUNG(2SPACE)+
      "                                      <094>
4002 PRINT"(2DOWN)-1- NACH NUMMER"           <191>
4011 PRINT"(DOWN)-2- NACH NAMEN"             <082>
4020 GOSUB 1:IF A$="(F1)"THEN 100            <232>
4030 IF A$="1"THEN INPUT"(DOWN)NUMMER
      ";Y:GOTO 4499                          <243>
4040 PRINT"(CLR)";PT$:PRINT"(3DOWN)NA
      ME":GOSUB 4:GOSUB 5:B$=A$:IF F=1
      THEN 100                               <013>
4050 GOSUB 4100:IF F<>999 THEN 4499          <159>
4060 GOTO 4230                               <086>
4100 CLOSE 2:OPEN 2,FA,15:OPEN 5,FA,2
      ,"AD,S,R":GOSUB 34000:IF ER<>0 T
      HEN 100                                <238>
4110 GOSUB 34100:INPUT#5,A                   <022>
4120 IF LEFT$(A$,LEN(B$))=B$THEN CLOS
      E 2:CLOSE 5:Y=A:RETURN                 <089>
4150 IF(ST AND 64)=64 THEN CLOSE 2:CL
      OSE 5:GOTO 4200                        <021>
4160 GOTO 4110                               <108>
4200 F=999:PRINT"(CLR)";PT$                  <123>
4210 PRINT"(2DOWN)DATENSATZ ";B$             <082>
4220 PRINT"NICHT VORHANDEN.":RETURN          <098>
4230 PRINT"(2DOWN)BITTE [F1] DRUECKEN
      ":GOSUB 1:GOTO 100                     <137>
4499 GOSUB 4500:GOTO 4560                    <135>
4500 CLOSE 2:CLOSE 5                         <205>
4501 HB=INT(Y/256):LB=Y-HB*256               <141>
4510 OPEN 2,FA,15:OPEN 5,FA,2,"ASALEM
      ,L,"+CHR$(254)                         <163>
4520 PRINT#2,"P"+CHR$(2)+CHR$(LB)+CHR
      $(HB)+CHR$(1)                          <209>
4522 INPUT#2,EN                              <108>
4525 IF EN<>0 THEN PRINT"(RVSON)ERROR
      FALSCHE DISKETTE !!!":RETURN           <093>
4530 B=20:GOSUB 4700:GOSUB 4750:V$(1)
      =B$                                    <234>
4531 B=20:GOSUB 4700:GOSUB 4750:V$(2)
      =B$                                    <043>
4532 B=20:GOSUB 4700:GOSUB 4750:V$(3)
      =B$                                    <108>
4533 B=4:GOSUB 4700:V$(4)=B$                 <105>
4534 B=20:GOSUB 4700:GOSUB 4750:V$(4)
      =V$(4)+" "+B$                          <220>
4536 B=16:GOSUB 4700:V$(5)=B$                <183>
4537 B=2:GOSUB 4700:V$(6)=B$+".":GOSU
      B 4700:V$(6)=V$(6)+B$+".":B=4:GO
      SUB 4700                               <043>
4538 V$(6)=V$(6)+B$:B=10:GOSUB 4700:V
      $(7)=B$                                <255>
4539 B=2:GOSUB 4700:V$(8)=B$                 <081>
4542 B=10:GOSUB 4700:V$(9)=B$                <001>
4544 B=10:GOSUB 4700:V$(10)=B$               <204>
4546 B=20:GOSUB 4700:GOSUB 4750:V$(11
      )=B$                                   <240>
4548 B=20:GOSUB 4700:V$(12)=B$               <228>
4549 B=20:GOSUB 4700:GOSUB 4750:V$(13
      )=B$                                   <244>
4550 B=20:GOSUB 4700:GOSUB 4750:V$(14
      )=B$                                   <118>
4555 CLOSE 2:CLOSE 5:RETURN                  <175>
4560 PRINT"(CLR)";PT$:PRINT"DATENSATZ
      ";V$(2)                                <174>
4561 IF BA$<>V$(8)AND PR=0 THEN V$(7)
      ="PASSWORT GESCHUETZT"                 <108>
4565 PRINT:FOR T=1 TO 14:PRINT V$(T):
      NEXT:PRINT"(DOWN)BITTE [F1] DRUE
      CKEN":GOSUB 1:GOTO 100                 <196>
4599 GOTO 100                                <209>
4700 POKE 917,B-1:POKE 915,5:SYS 5018
      2,B$                                   <245>
4740 IF RIGHT$(B$,1)=" "THEN B$=LEFT$
      (B$,LEN(B$)-1):GOTO 4740               <235>
4749 RETURN                                  <235>
4750 A$=LEFT$(B$,1):IF ASC(A$)<65 OR
      ASC(A$)>90 THEN RETURN                 <061>
4760 B$=RIGHT$(B$,LEN(B$)-1):B$=CHR$(
      ASC(A$)+32)+B$:RETURN                  <070>
5000 GOTO 5500                               <226>
5001 PRINT"(CLR)";PT$:PRINT"(2DOWN)LA
      DEN VON DISKETTE"                      <225>
5010 PRINT"(2DOWN,3SPACE)-1- TEXT LOE
      SCHEN UND NEUEN(17SPACE)TEXT LAD
      EN"                                    <166>
5020 PRINT"(DOWN,3SPACE)-2- HINTER AL
      TEN TEXT LADEN"                        <035>
5021 PRINT"(DOWN,3SPACE)-3- LISTINGS
      LADEN"                                 <167>
5025 PRINT"(DOWN)BEFEHL";                    <229>
5030 GOSUB 1                                 <139>
5040 IF A$="(F1)"THEN 100                    <169>
5041 B1$="ANSEHEN [F3]":B2$="AENDERN
      [2]"                                   <244>
5042 IF A$="1"OR A$="3"THEN GOSUB 130
      5:GOTO 5050                            <193>
5043 IF A$="2"THEN 5050                      <105>
5045 GOTO 5030                               <031>
5050 PRINT"-";A$;"-":I=0:IF A$="3"THE
      N I=999                                <066>
5060 PRINT"(CLR,5DOWN)NAME DER DATEI"
      :A$="":GOSUB 9:T1$=A$:B$=A$:IF F
      =1 THEN 100                            <015>
5061 IF LEFT$(A$,1)<>"@"THEN 5066            <169>
5062 A$=RIGHT$(A$,LEN(A$)-1):T=VAL(A$
      ):A$=V$(30+T)                          <241>
5063 IF T>B OR B=0 THEN 5060                 <155>
5064 GOTO 5100                               <226>
5066 PRINT"(HOME,5DOWN)AUFZEICHNUNGSD
      ATUM":A$="??????":GOSUB 9              <255>
5070 IF PR<>1 THEN A$=">"+BA$+A$+B$          <135>
5071 IF PR=1 THEN A$=B$                      <230>
5100 REM                                     <080>
5102 OPEN 2,FA,15,"I0"                       <145>
5103 GOSUB 34000                             <081>
5104 OPEN 5,FA,5,"0:"+A$+",S,R"              <181>
5105 AX=0:GOSUB 34000:IF EN=62 THEN 1
      00                                     <237>
5106 IF I=999 THEN 5200                      <041>
5107 INPUT#5,AX:FOR T=1 TO AX                <245>
5110 REM                                     <090>
5111 GOSUB 34100:A$(T+AN)=A$:IF A$(T+
      AN)=">"THEN A$(T+AN)=" "               <226>
5112 GOSUB 34000                             <090>
5113 GET A$:IF A$="(F1)"THEN CLOSE 2:
      CLOSE 5:GOTO 100                       <080>
5114 NEXT                                    <042>
5115 AN=AN+AX                                <196>
5120 CLOSE 2:CLOSE 5:GOTO 100                <136>
5200 REM                                     <182>
5210 AN=AN+1                                 <101>
5211 GET A$:IF A$="(F1)"THEN CLOSE 2:
      CLOSE 5:GOTO 100                       <180>
5220 GOSUB 34100:A$(AN)=A$:IF ST<>64
      THEN 5210                              <127>
5230 CLOSE 2:CLOSE 5:GOTO 100                <246>
5400 PRINT"(CLR)";PT$:PRINT"(2DOWN)VE
      RGLEICHEN VON DISKETTE"                <211>
5401 A$=T1$:PRINT"NAME":GOSUB 9:IF F=
      1 THEN 100                             <030>
5402 OPEN 2,FA,15,"I0":GOSUB 70              <153>
5403 GOSUB 34000:IF PR<>1 THEN A$=">"
      +BA$+"??????"+A$                       <073>
5404 OPEN 5,FA,5,"0:"+A$+",S,R"              <229>
5405 GOSUB 34000:IF EN<>0 THEN 100           <252>
5407 INPUT#5,AX:IF B<>0 AND AX<>B-V T
      HEN PRINT"ZEILENANZAHL FALSCH"         <204>
5408 FOR T=1 TO AX                           <033>
5410 GOSUB 34100:IF A$=">"THEN A$=" "        <226>
5412 PRINT A$:POKE 198,0:WAIT 198,1:G
      ET A$:IF A$="(F1)"THEN 5420            <077>
5413 GOSUB 34000                             <139>
5415 NEXT                                    <091>
5420 CLOSE 2:CLOSE 5:PRINT"TASTE DRUE
      CKEN":GOSUB 1:GOTO 100                 <237>
5500 PRINT"(CLR)";PT$                        <044>
5501 PRINT"(RVSON,SPACE)+ MENU DISKET
      TEN-OPERATIONEN(9SPACE)+ "             <133>
5502 PRINT"(DOWN,3SPACE)LADEN(5SPACE)[
      4]"                                    <051>
5503 PRINT" SPEICHERN [5]"                   <058>
5510 REM                                     <238>
5520 PRINT"(DOWN,2SPACE)-0- MENU"            <229>
5521 PRINT"(2SPACE)-1- TEXT VERGLEICH
      EN"                                    <055>
5525 PRINT"(2SPACE)-2- TEXT UMBENENNE
      N"                                     <243>
5527 PRINT"(2SPACE)-3- TEXT LOESCHEN"        <250>
5529 PRINT"(2SPACE)-4- TEXT LADEN"           <037>
5530 PRINT"(2SPACE)-5- TEXT SPEICHERN
      "                                      <151>
5532 PRINT"(2SPACE)-6- INHALTSVERZEIC
      HNIS"                                  <159>
5534 IF PR=1 THEN PRINT"(2SPACE)-7- D
      ISK FORMATIEREN"                       <174>
5536 IF PR=1 THEN PRINT"(2SPACE)-8- D
      ISK NAME + ID AENDERN"                 <023>
5550 GOSUB 1                                 <151>
5551 IF A$="0"OR A$="(F1)"THEN 100           <029>
5552 IF VAL(A$)=0 THEN 5500                  <059>
5553 ON VAL(A$)GOTO 5400,5650,5700,50
      01,6000,5800,5600,5900                 <253>
5555 RETURN                                  <023>
5600 PRINT"(CLR)";PT$:PA$="":IF PR=0
      THEN 100                               <064>
5610 PRINT"(RVSON,SPACE)D I S K E T T
      E(2SPACE)F O R M A T I E R E N
      "                                      <209>
5630 A$="DISKETTEN NAME":GOSUB 9:B$=A
      $:IF F=1 THEN 100                      <009>
5631 PRINT"(HOME,4DOWN)SCHUETZEN?":A$
      ="N":GOSUB 9:IF A$="J"THEN INPUT
      "PASSWORT";PA$                         <155>
5633 IF A$="J"AND LEN(PA$)<10 THEN PA
      $=PA$+CHR$(0):GOTO 5633                <041>
5634 A$="ID"                                 <089>
5635 PRINT"(HOME,4DOWN)DISKETTEN ID(2
      SPACE)":GOSUB 9:ID$=A$:A$=LEFT$(
      B$+"................",16)              <166>
5640 ID$=LEFT$(ID$+"TE",2)                   <081>
5641 GOSUB 3                                 <004>
5642 OPEN 2,FA,15,"N:"+A$+","+ID$:CLO
      SE 2:IF PA$=""THEN GOTO 100            <034>
5643 OPEN 15,FA,15,PR$:OPEN 2,FA,2,"#
      "                                      <143>
5644 PRINT#15,"B-R 2 0 18 0":PRINT#15
      ,"B-P 2 164"                           <169>
5645 PRINT#2,CHR$(0)+CHR$(0)+CHR$(0):
      PRINT#15,"B-P 2 180":PRINT#2,PA$       <172>
5646 PRINT#15,"B-P 2 2":PRINT#2,"0";         <229>
5647 PRINT#15,"B-W 2 0 18 0":CLOSE 2:
      CLOSE 15:GOTO 100                      <220>
5650 PRINT"(CLR)";PT$:PRINT"(RVSON,2S
      PACE)+(2SPACE)T E X T(3SPACE)U M
      B E N E N N E N(2SPACE)+(3SPACE
      )"                                     <169>
5651 PRINT"(HOME,4DOWN)ALTER NAME":A$
      ="":GOSUB 9:IF F=1 THEN 100            <013>
5652 B$=A$:A$="TTMMJJ":PRINT"(HOME,4D
      OWN)DATUM":GOSUB 9                     <058>
5655 C$=A$:PRINT"(HOME,4DOWN)NEUER NA
      ME":GOSUB 9                            <173>
5656 IF PR=1 THEN GOSUB 3:A$=">"+A$:B
      $=">"+B$                               <191>
5657 IF PR=0 THEN GOSUB 3:A$=">"+BA$+
      DA$+A$:B$=">"+BA$+C$+B$                <223>
5660 OPEN 2,8,15,PR$                         <101>
5670 PRINT#2,"R:"+A$+"="+B$:GOSUB 340
      00:CLOSE 2:GOTO 100                    <074>
5700 PRINT"(CLR)";PT$:PRINT"(RVSON,4S
      PACE)+(2SPACE)T E X T(3SPACE)L O
      E S C H E N(2SPACE)+(5SPACE)"          <129>
5715 PRINT"(HOME,4DOWN)NAME":GOSUB 9:
      B$=A$:IF F=1 THEN 100                  <156>
5716 A$="??????":PRINT"(HOME,4DOWN)DA
      TUM(6SPACE)":GOSUB 9:GOSUB 3           <007>
5717 IF PR<>1 THEN A$=">"+BA$+A$+B$          <020>
5718 IF PR=1 THEN A$=">"+B$                  <070>
5719 OPEN 2,FA,15,PR$                        <248>
5720 PRINT#2,"S0:"+A$:CLOSE 2:GOTO 10
      0                                      <250>
5800 CLOSE 15:CLOSE 2:OPEN 15,FA,15,"
      I":PRINT"(CLR)";PT$                    <096>
5801 PRINT"(CLR)";PT$:PRINT"(RVSON,2S
      PACE)+ INHALTSVERZEICHNIS DER DI
      SKETTE(2SPACE)+(2SPACE,3DOWN)"         <191>
5810 OPEN 2,FA,2,"#"                         <042>
5811 PRINT#15,"B-R 2 0 18 0":PRINT#15
      ,"B-P 2 165"                           <096>
5812 GET#2,A$:IF A$<>""THEN 5816             <231>
5813 PRINT#15,"B-P 2 180":PA$="":FOR
      T=1 TO 10:GET#2,A$:PA$=PA$+A$:NE
      XT                                     <231>
5814 PRINT"DIE DISKETTE IST GESCHUETZ
      T: GEBEN SIE(2SPACE)DAS PASSWORT
      EIN"                                   <240>
5815 INPUT A$:IF A$<>PA$THEN CLOSE 2:
      CLOSE 15:GOTO 100                      <135>
5816 B$="":X=999:GOSUB 5910:X=0:B$="(
      F3)"                                   <164>
5817 IF A$(2)="INHALT"THEN AN=AN+2:A$
      (AN)=ID$+" "+N$:AN=AN+1                <064>
5820 T=18:S=1:B=0                            <094>
5821 GOSUB 5898                              <128>
5830 PRINT#15,"U1 2 0";T;S                   <001>
5840 V=0                                     <237>
5841 GET#2,X$:X$=X$+CHR$(0):T=ASC(X$)        <178>
5842 GET#2,X$:X$=X$+CHR$(0):S=ASC(X$)        <163>
5845 FOR X=0 TO 7:PRINT#15,"B-P 2";X*
      32+2                                   <228>
5846 GET#2,X$,A$:Z=ASC(X$+CHR$(0)):IF
      Z<>B*16+1 AND PR=0 THEN 5855           <054>
5847 GOSUB 5890                              <152>
5848 IF B=10 THEN B=0:PRINT"(DOWN)BIT
      TE F3 DRUECKEN(DOWN)":WAIT 198,1
      :GET B$:GOSUB 5898                     <153>
5849 IF B$="(F1)"THEN 100                    <224>
5850 X$="(WHITE)SEQ ":IF Z<>0 AND PR=
      1 THEN GOSUB 5870                      <076>
5851 IF PR=0 AND Z<>0 THEN PRINT"(WHI
      TE,3SPACE)";B,"(LEFT)";FF$:V$(30
      +B)=C$                                 <002>
5852 IF PR=1 THEN PRINT"(WHITE,2SPACE
      )";X$,FF$                              <147>
5855 NEXT X                                  <215>
5856 IF T<>0 THEN 5830                       <030>
5860 PRINT"(2DOWN)BITTE F1 DRUECKEN":
      GOSUB 1                                <249>
5865 GOTO 100                                <205>
5870 IF Z=130 THEN X$="PRG "                 <080>
5871 IF Z=129 THEN X$="SEQ "                 <183>
5872 IF Z=194 THEN X$="PRG<"                 <052>
```

Listing 1. »Hypra-Text«-Hauptprogramm. Bitte die Eingabehinweise auf Seite 54 beachten.



```
5873 IF Z=131 THEN X$="USR "                <057>
5874 IF Z=193 THEN X$="SEQ<"                <215>
5875 IF Z=195 THEN X$="USR<"                <029>
5880 IF A$(1)="P"AND Z<>130 AND Z<>19
     4 THEN RETURN                          <122>
5881 IF A$(2)<>"INHALT"THEN RETURN          <185>
5885 AN=AN+1:A$(AN)=ID$+" "+X$+FF$:RE
     TURN                                   <241>
5888 GOTO 100                               <230>
5890 IF PR=1 THEN 5897                      <188>
5891 GET#2,A$,A$:IF A$<>">"THEN Z=0:R
     ETURN                                  <137>
5892 GET#2,A$,B$:FF$=A$+B$:IF FF$<>BA
     $AND BA$<>"--"THEN Z=0:RETURN          <239>
5893 C$=">"+A$+B$:GET#2,A$,B$:FF$=FF$
     +"{2SPACE}"+A$+B$+".":C$=C$+A$+B
     $                                      <086>
5894 GET#2,A$,B$:FF$=FF$+A$+B$+".":C$
     =C$+A$+B$                              <247>
5895 GET#2,A$,B$:FF$=FF$+A$+B$+"{2SPA
     CE,SHIFT-SPACE}":C$=C$+A$+B$           <140>
5896 FOR Y=1 TO 7:GET#2,A$:C$=C$+A$:F
     F$=FF$+A$:NEXT Y:B=B+1:RETURN          <047>
5897 FF$="":GET#2,A$:FOR Y=0 TO 15:GE
     T#2,A$:FF$=FF$+A$:NEXT:B=B+1:RET
     URN                                    <123>
5898 PRINT"{CLR}";PT$:PRINT"DISK: ";N
     $;"{2SPACE}ID:";ID$;"{DOWN}"           <172>
5899 PRINT"{4SPACE}NR.{2SPACE}HER{2SP
     ACE}DATUM{5SPACE}NAME{DOWN}":RET
     URN                                    <143>
5900 IF PR=0 THEN 100                       <149>
5901 PRINT"{CLR}";PT$:PRINT"{RVSON}AE
     NDERN DES DISKNAMENS{2SPACE}ODER
      DER DISK ID"                          <057>
5902 PRINT"{DOWN}DIE ZU BENDERNDE DIS
     KETTE EINLEGEN UND":PRINT"{2SPAC
     E}[F3]{2SPACE}DRUECKEN"                <127>
5903 POKE 198,0:GOSUB 1:IF A$<>"{F3}"
     THEN 100                               <113>
5908 B$=""                                  <099>
5909 OPEN 15,8,15,"I":PRINT#15,PR$:OP
     EN 2,8,2,"#"                           <213>
5910 PRINT#15,"B-R 2 0 18 0":PRINT#15
     ,"B-P 2 144":FOR T=1 TO 20:GET#2
     ,A$:B$=B$+A$:NEXT                      <010>
5920 N$=LEFT$(B$,16):ID$=RIGHT$(B$,2)       <135>
5921 IF X=999 THEN RETURN                   <063>
5925 PRINT"NAME{2SPACE}";N$:INPUT"{UP
     }NAME";N$                              <069>
5927 PRINT"ID{4SPACE}";ID$:INPUT"{UP}
     ID{2SPACE}";ID$                        <065>
5930 N$=N$+"{16SHIFT-SPACE}":ID$=ID$+
     "++":N$=LEFT$(N$,16):ID$=LEFT$(I
     D$,2)                                  <153>
5940 PRINT#15,"B-P 2 144":PRINT#2,N$;
     ID$;:PRINT#15,"B-W 2 0 18 0"           <171>
5945 CLOSE 2:CLOSE 15                       <023>
5950 INPUT"PASSWORT";PA$:IF PA$="0"TH
     EN 100                                 <163>
5960 IF LEN(PA$)<10 THEN PA$=PA$+CHR$
     (0):GOTO 5960                          <120>
5970 GOTO 5643                              <004>
6000 PRINT"{CLR}";PT$:PRINT"{RVSON}+
     + T E X T{4SPACE}A U F Z E I C H
      N E N + +"                            <201>
6001 PRINT"{DOWN}TEXTNAME: ";T1$            <163>
6002 PRINT"{2DOWN,SPACE}[F1]=ENDE{3SP
     ACE}[=] = TEXTNAME{DOWN}"              <128>
6004 A$="=":GOSUB 9:IF F=1 THEN 100         <175>
6005 IF A$="="THEN A$=T1$:IF T1$=""TH
     EN PRINT"NAME FEHLT":GOTO 6004         <150>
6006 T1$=A$:B$=">"+BA$+"??????"+A$          <230>
6007 PRINT"{DOWN}ALLE VORGAENGER LOES
     CHEN (J/N)?":GOSUB 1                   <230>
6008 IF A$="J"THEN OPEN 2,8,15,"S:"+B
     $:CLOSE 2                              <027>
6009 B1$="DRUCKEN [3]":B2$="ENDE [F5]
     "                                      <223>
6015 A$=T1$:GOSUB 70                        <152>
6016 T1$=A$:OPEN 2,FA,15,"I0":A$=">"+
     BA$+DA$+A$:PRINT#2,PR$                 <073>
6017 GOSUB 34000                            <235>
6018 OPEN 5,FA,5,"@0:"+A$+",S,W"            <077>
6021 GOSUB 34000                            <239>
6022 AX=B-V:PRINT#5,STR$(AX)                <153>
6024 FOR T=V TO B:IF A$(T)=""OR A$(T)
     =" "THEN A$(T)=">"                     <149>
6030 PRINT#5,A$(T):IF A$(T)=">"THEN A
     $(T)=""                                <114>
6031 GOSUB 34000                            <249>
6032 NEXT:CLOSE 2:CLOSE 5:GOTO 100          <040>
7000 T=0:X=0                                <093>
7002 IF X<0 THEN X=0                        <232>
7003 GOSUB 7030:FOR T=1 TO 20               <042>
7004 PRINT"{RVSON}";RIGHT$(STR$(T+X*2
     0),1);"{RVOFF}";A$(T+X*20)             <074>
7006 NEXT T                                 <064>
7007 PRINT"{DOWN}F1=ENDE{3SPACE}F3=SE
     ITE +{3SPACE}F5=SEITE - "              <201>
7008 B1$="AENDERN{3SPACE}[2]"               <194>
7009 B2$="SPEICHERN [5]":B3$="HILFE{5
     SPACE}[F7]"                            <117>
7010 GOSUB 1                                <087>
7011 IF A$="{F1}"THEN 100                   <108>
7012 IF A$="{F5}"THEN X=X-1:IF X>=0 T
     HEN 7002                               <133>
7013 IF A$="{F3}"THEN X=X+1:IF X*20<A
     N THEN 7002                            <199>
7014 IF X*20>=AN THEN X=0:GOTO 7002         <137>
7015 IF X<0 THEN X=INT(AN/20):GOTO 70
     02                                     <248>
7020 GOTO 7010                              <182>
7030 PRINT"{CLR,RVSON,SPACE}+T E X T{
     2SPACE}6 4 +BILDSCHIRM+ZEILE{8SP
     ACE}"                                  <026>
7031 PRINT"{HOME,RVSON,SPACE}+T E X T
     {2SPACE}6 4 +BILDSCHIRM+ZEILE";X
     *20;"-"                                <253>
7032 RETURN                                 <232>
7100 PRINT"{CLR}";PT$:PRINT"{RVSON,SP
     ACE}+ +{6SPACE}BILDSCHIRMAUSGABE
     {9SPACE}+ + "                          <106>
7105 INPUT"{2DOWN}AB ZEILE";X:IF X<1
     OR X>AN THEN X=1                       <083>
7106 X=INT(X/20):GOTO 7002                  <021>
8000 PRINT"{CLR}";PT$                       <004>
8002 G=0                                    <053>
8006 PRINT"{DOWN,SPACE}-1- NEUSTART D
     ES PROGRAMMS"                          <122>
8007 PRINT"{DOWN,SPACE}-2- INFO UEBER
      SPEICHERPLATZ"                        <084>
8008 PRINT"{DOWN,SPACE}-3- TON AUS"         <017>
8009 PRINT"{DOWN,SPACE}-4- TON AN"          <052>
8010 GOSUB 1                                <071>
8011 IF A$="1"THEN 10000                    <068>
8013 IF A$="0"OR A$="{F1}"THEN 100          <207>
8014 IF A$="1"THEN RUN                      <023>
8015 IF A$="2"THEN PRINT FRE(9);"ZEIC
     HEN FREI":PRINT"{DOWN,SPACE}TAST
     E DRUECKEN":GOSUB 1:GOTO 1             <252>
8016 IF A$="{HOME}"THEN INPUT"CODE";A
     $:IF A$="SUSI"THEN 55555               <032>
8017 IF A$="{CTRL-P}"AND(BA$="--"OR B
     A$="OF")THEN PR=1:PRINT"{3DOWN}P
     ROG OK":GOSUB 1:GOTO 100               <115>
8018 IF A$="{CTRL-O}"THEN PR=0:PRINT"
     {3DOWN}PROGRAMMIERER AUS":GOSUB
     1:GOTO 100                             <188>
8019 IF A$="3"THEN T9%=1:PRINT"{DOWN}
     TON AUS":GOTO 100                      <140>
8020 IF A$="4"THEN T9%=0                    <067>
8021 IF A$="{WHITE}"AND PR=1 THEN END       <059>
8099 GOTO 100                               <153>
9400 REM*** FILL ***                        <042>
9401 PRINT"{CLR}";PT$:PRINT"{HOME,2DO
     WN,RVSON,SPACE}* * TEXTPROGRAMM
     * * AUSFUELLEN{2SPACE}* *{3SPACE
     }"                                     <006>
9407 F2=0:PRINT"NEBEN DEN TEXT [J/N]"
     :GOSUB 1:IF A$="J"THEN F2=1            <254>
9408 IF A$="{F1}"THEN 100                   <219>
9409 IF F2=1 THEN PRINT"LINKS/RECHTS
     [L/R]":GOSUB 1:G=1:IF A$="R"THEN
      G=2                                   <227>
9410 GOSUB 70                               <252>
9421 INPUT"SCHRITTWEITE";S:IF S<=0 TH
     EN 9421                                <182>
9429 PRINT"{CLR,4DOWN}WAS ?"                <081>
9430 A$="":GOSUB 9:IF F=1 THEN 100          <197>
9440 IF B>AN THEN AN=B                      <202>
9450 FOR T=V TO B STEP S                    <104>
9452 IF F2=0 THEN A$(T)=A$                  <001>
9453 IF F2=1 AND G=1 THEN A$(T)=A$+A$
     (T)                                    <243>
9454 IF F2=1 AND G=2 THEN A$(T)=A$(T)
     +A$                                    <151>
9455 NEXT:GOTO 7000                         <228>
9689 RETURN                                 <093>
9700 PRINT"{CLR,RVSON}";PT$;"{DOWN}"        <114>
9710 PRINT"{DOWN,2SPACE}1{2SPACE}VARI
     ABLEN ZEIGEN"                          <168>
9712 PRINT"{DOWN,2SPACE}2{2SPACE}VARI
     ABLEN EINGEBEN"                        <149>
9713 PRINT"{DOWN,2SPACE}3{2SPACE}VARI
     ABLEN AN"                              <170>
9714 PRINT"{DOWN,2SPACE}4{2SPACE}VARI
     ABLEN AUS"                             <188>
9715 PRINT"{DOWN,2SPACE}5{2SPACE}VARI
     ABLEN ZU TEXT"                         <069>
9717 PRINT"{DOWN,2SPACE}6{2SPACE}VARI
     ABLEN ALS TEXT 1-50"                   <249>
9718 PRINT"{DOWN,2SPACE}7{2SPACE}TEXT
      ALS VARIABLEN 1-50"                   <198>
9719 PRINT"{DOWN,2SPACE}8{2SPACE}VARI
     ABLEN AUS DATEIVERWALTUNG"             <230>
9750 GOSUB 1                                <033>
9755 IF A$="0"OR A$="{F1}"THEN 100          <171>
9760 ON VAL(A$)GOTO 9800,9900,9780,97
     85,9850,9790,9795,4000                 <109>
9770 GOTO 9750                              <138>
9780 V9=1:PRINT"VARIABLEN AN":FOR T=1
      TO 999:NEXT:GOTO 9700                 <091>
9785 V9=0:PRINT"VARIABLEN AUS":FOR T=
     1 TO 999:NEXT:GOTO 100                 <214>
9790 FOR T=1 TO 50:A$(T)=V$(T):AN=50:
     NEXT:GOTO 100                          <149>
9795 FOR T=1 TO 50:V$(T)=A$(T):NEXT:G
     OTO 100                                <165>
9800 PRINT"{CLR}";PT$                       <026>
9801 FOR T=1 TO 50:PRINT T,V$(T)            <076>
9810 IF T/20=INT(T/20)THEN GOSUB 1:PR
     INT"{CLR}";PT$                         <224>
9820 NEXT:GOSUB 1:GOTO 100                  <036>
9850 GOSUB 3:V9=1                           <218>
9860 FOR T=1 TO AN:A$=A$(T):GOSUB 303
     00:A$(T)=A$:NEXT:V9=0:GOTO 100         <111>
9900 PRINT"{CLR}";PT$:PRINT"{DOWN,6SP
     ACE}+ + VARIABLEN EINGABE + +{4D
     OWN}"                                  <149>
9901 INPUT"VARIABLEN NR.";A:IF A>40 T
     HEN 9900                               <126>
9910 A$="":GOSUB 9:V$(A)=A$                 <196>
9920 PRINT:PRINT"NOCH EINE? [J/N]"          <161>
9930 GOSUB 1                                <213>
9940 IF A$="N"THEN 100                      <088>
9950 GOTO 9900                              <190>
10000 PRINT"{CLR}";PT$:B$=A$                <216>
10010 PRINT"{2DOWN}ACHTUNG !!! TEXT U
      NBEDINGT SPEICHERN"                   <023>
10020 PRINT"{DOWN}IST DER TEXT SCHON
      GESPEICHERT? [J/N]"                   <083>
10030 GOSUB 1:IF A$<>"J"THEN 6000           <236>
10040 A$=B$:POKE 930,0:GOTO 8012            <162>
20000 PRINT"{CLR}";PT$:PRINT"{DOWN,RV
      SON,SPACE}+ + TEXTPROGRAMM + +
      R E C H N E N{2SPACE}+ +"             <092>
20009 PRINT"{DOWN,SPACE}[E]= ENDE"          <114>
20010 PRINT"{DOWN,SPACE}[C]= SPEICHER
       LOESCHEN"                            <106>
20011 PRINT"{DOWN,SPACE}[M]= MEHRWERT
      STEUER EINGEBEN"                      <183>
20013 PRINT"{DOWN,SPACE}[+] [-] [*] [
      :] = RECHENOPERATIONEN"               <200>
20015 RR=INT((RR*100)+0.5)/100              <075>
20019 PRINT"{2DOWN,RVSON}";RR;"{RVOFF
      }";                                   <168>
20020 GOSUB 1                               <143>
20030 IF A$="C"THEN RR=0:GOTO 20000         <120>
20031 IF A$="E"OR A$="{F1}"THEN 100         <169>
20032 IF A$="M"THEN INPUT"STEUER IN %
      {3SPACE}14{4LEFT}";MW:GOTO 2000
      0                                     <235>
20034 IF A$<>"+"AND A$<>"-"AND A$<>"*
      "AND A$<>":"THEN 20020                <234>
20040 PRINT A$;"{2SPACE}";:INPUT X          <029>
20050 IF A$=":"AND X=0 THEN PRINT"FEH
      LER... NICHT DURCH 0 DIVIDIEREN
      ":GOTO 20020                          <227>
20060 IF A$=":"THEN RR=RR/X                 <240>
20061 IF A$="*"THEN RR=RR*X                 <105>
20062 IF A$="+"THEN RR=RR+X                 <233>
20063 IF A$="-"THEN RR=RR-X                 <107>
20064 GOTO 20000                            <224>
22000 IF PEEK(1022)=0 THEN GOSUB 2210
      0                                     <155>
22001 IF PEEK(1022)=0 THEN POKE 1022,
      1:LOAD"ZEICHEN B",8,1                 <157>
22002 IF PEEK(1022)=250 THEN POKE 102
      2,240:LOAD"ZEICHEN DIN",8,1           <245>
22003 IF PEEK(1022)=1 THEN POKE 53272
      ,11:POKE 56576,PEEK(56576)AND 2
      52:POKE 648,192                       <123>
22004 IF PEEK(1022)-240 THEN POKE 532
      72,11:POKE 56576,PEEK(56576)AND
       252:POKE 648,192                     <226>
22005 IF PEEK(1022)=1 THEN POKE 1022,
      2:LOAD"ZEICHEN D",8,1                 <050>
22006 PRINT"{CLR}":IF PEEK(1022)=240
      THEN POKE 1022,199:LOAD"ZEICHEN
       DDIN",8,1                            <027>
22007 IF PEEK(1022)=2 THEN POKE 1022,
      3:LOAD"PROFIP V1.0",8,1               <087>
22008 IF PEEK(1022)=199 THEN POKE 102
      2,200:LOAD"PROFIP V1.0",8,1           <233>
22009 IF PEEK(1022)=111 THEN POKE 532
      72,11:POKE 56576,PEEK(56576)AND
       252:POKE 648,192:END                 <054>
22010 GOTO 11                               <102>
22100 PRINT CHR$(14);"{CLR,RVSON,SPAC
      E}TEXTVERARBEITUNGSPROGRAMM{6SP
      ACE}O.FENDT "                         <202>
22105 PRINT"{2DOWN}WAS FUER EINE TAST
      ATURBELEGUNG?"                        <243>
22106 INPUT"{2DOWN,SPACE}[DIN] - [CBM
      ]{3RIGHT}DIN{5LEFT}";A$               <084>
22107 IF A$="DIN"THEN POKE 1022,250         <237>
22108 GOSUB 3:RETURN                        <075>
30000 G9=G8-6:GOSUB 30700:XT=0:IF LEN
      (A$)>M8 THEN A$=LEFT$(A$,M8)          <068>
30001 POKE 922,AM:CF=1:IF U9=2 THEN U
      9=0:CP=LEN(R$)+CB:GOTO 30020          <105>
30002 IF CC=999 THEN CP=LEN(A$(G8)):C
      C=0:GOTO 30020                        <097>
30010 CC=0:X=0:CP=CB                        <150>
30020 IF CP=0 THEN CP=1                     <230>
30021 POKE 908,CP-1:POKE 918,M8-(M8/5
      ):POKE 920,11:POKE 917,M8:POKE
      924,0                                 <232>
30040 SYS 50176,A$,A$ :F=PEEK(913):IF
       A$="> "THEN A$=""                    <218>
30045 GOSUB 31                              <041>
30053 IF F=9 THEN GOSUB 30550:GOTO 30
      020                                   <222>
30054 IF F=15 THEN A$="":A$(G8)="":GO
      TO 30000                              <124>
30055 IF F=11 THEN GOSUB 30400:A$(G8)
      =A$                                   <072>
30056 IF F=20 THEN CC=999:F=5               <229>
30057 IF F=16 THEN F=0:IF RIGHT$(A$,1
      )<>"F"THEN A$=A$+"F"                  <003>
30099 GOSUB 30300:GOSUB 30400:PRINT:R
      ETURN                                 <052>
30300 REM ** VARIABLEN TEST **              <186>
30301 IF V9=0 THEN RETURN                   <127>
30310 FOR B=1 TO LEN(A$)-2                  <072>
30320 IF MID$(A$,B,1)<>"@"THEN NEXT:R
      ETURN                                 <157>
30330 X-VAL(MID$("00"+A$,B+1,2)):IF X
      >50 THEN X=0                          <114>
30340 A$=LEFT$(A$,B-1)+V$(X)+RIGHT$(A
      $,LEN(A$)-B-2):GOTO 30310             <088>
30350 NEXT:RETURN                           <061>
30400 IF LEN(A$)<M8 THEN RETURN             <145>
30401 IF LEN(A$)>80 THEN 30440              <238>
30403 CC=999:FOR I=M8 TO 1 STEP-1           <105>
30410 IF MID$(A$,I,1)<>" "THEN NEXT I
      :GOTO 30440                           <021>
30420 R$=RIGHT$(A$,LEN(A$)-I):A$=LEFT
      $(A$,I-1):U9=1                        <010>
30421 IF RIGHT$(R$,1)=" "THEN R$=LEFT
      $(R$,LEN(R$)-1):GOTO 30421            <174>
30430 IF LEN(A$)>=M8 THEN NEXT:GOTO 3
      0440                                  <140>
30431 RETURN                                <005>
30440 GOSUB 95:PRINT"{DOWN}>> ZEILE Z
      U LANG << FEHLER !!":POKE 198,0       <044>
30441 A$=LEFT$(A$,M8-1)                     <133>
30442 GOSUB 30601:IF PEEK(198)<1 THEN
       30442                                <134>
30443 POKE 198,0:RETURN                     <210>
30550 IF CP+1>LEN(T7$)OR XT=1 THEN RE
      TURN                                  <136>
30560 X=0:FOR Z=CP+1 TO LEN(T7$):IF M
      ID$(T7$,Z,1)="+"AND X=0 THEN X=
      1:CP=Z                                <190>
30570 NEXT:IF X=0 THEN XT=1                 <010>
```

Listing 1. (Fortsetzung)

```
30580 IF LEN(A$)<CP THEN A$=A$+"{SHIF
      T-SPACE}":GOTO 30580              <245>
30590 RETURN                            <166>
30600 REM ** ERROR SYSTEM **            <074>
30601 IF T9%=1 THEN RETURN              <134>
30602 POKE 53280,1                      <172>
30603 POKE 54296,15:POKE 54277,25:POK
      E 54278,4*16+4                    <132>
30610 POKE 54273,29:POKE 54272,69:POK
      E 54276,17                        <047>
30619 FOR T=1 TO 200:NEXT               <080>
30620 POKE 53280,0:FOR T=1 TO 200:NEX
      T                                 <052>
30621 POKE 54276,0:POKE 54277,0:POKE
      54278,0:RETURN                    <228>
30700 REM   ** FENSTER **               <212>
30701 PRINT"{HOME}";PT$:PRINT"{2DOWN}
      ";T7$                             <134>
30702 GOSUB 80:GOSUB 30750              <012>
30705 PRINT"{HOME}":FOR I=0 TO 10:POK
      E 781,I+5:POKE 782,0:SYS 58636    <209>
30710 IF G9+I<=0 THEN GOSUB 90:GOTO 3
      0740                              <121>
30720 V=MB-LEN(A$(I+G9)):IF V<0 THEN
      V=0                               <138>
30730 PRINT A$(I+G9)+LEFT$(A6$,V)       <036>
30740 NEXT:GOSUB 90:RETURN              <149>
30750 PRINT K1$;LEFT$(A6$,(MB+2)-LEN(
      K1$))                             <073>
30751 PRINT K2$;LEFT$(A6$,(MB)-LEN(K2
      $))                               <022>
30752 PRINT K3$;LEFT$(A6$,(MB)-LEN(K3
      $))                               <014>
30753 PRINT K4$;LEFT$(A6$,(MB)-LEN(K4
      $)):RETURN                        <038>
31000 REM ***************************
      **                                <022>
31001 REM ***************************
      **                                <023>
31002 REM **      INITIALISIERUNG
      **                                <050>
31003 REM ***************************
      **                                <025>
31004 REM ***************************
      **                                <026>
31006 PT$="{WHITE,RVSON}HYPRA-TEXT V1
      .0{10SPACE}BY OLIVER FENDT{RVOF
      F}":ME=100:PRINT"{CLR}";PT$       <076>
31009 FU$="RECHNEN":MW=14               <016>
31010 DIM V$(50)                        <181>
31016 POKE 53281,0:POKE 53280,0:IF MB
      <=7 THEN MB=36                    <026>
31019 T7$="!----+-----+-----+-----+-----+
      -----+-----------"                <238>
31020 GOSUB 46000:B1$=STR$(MB)+" ZEIC
      HEN MODUS"                        <058>
31048 A6$="{67SPACE}"                   <033>
31060 B1$="LADEN{7SPACE}[5]"            <113>
31061 B2$="EINGABE{5SPACE}[F1]"         <039>
31062 B3$="HILFE{7SPACE}[F7]"           <172>
31063 GOSUB 1840:PR$="M-W"+CHR$(1)+CH
      R$(1)+CHR$(1)+"A"                 <152>
31150 PRINT"{CLR}";PT$:PRINT"{2DOWN}G
      UTEN TAG,";V$(1)                  <041>
31160 PRINT"{DOWN}BITTE GEBEN SIE DAS
       DATUM EIN."                      <039>
31199 RETURN                            <011>
33000 REM *** FORMBLATT ***             <052>
33001 PRINT"{CLR}";PT$:PRINT"{HOME,2D
      OWN,SPACE}+ M A S K E{2SPACE}AU
      SFUELLEN{2SPACE}+ VON MENU 1 +"
      :A=CB                             <213>
33004 B1$="SPEICHERN [5]":B2$="AENDER
      N [2]"                            <065>
33010 FOR PQ=1 TO AN                    <097>
33020 A$=A$(PQ):A$(PQ)=A$               <125>
33021 IF LEN(A$)=0 THEN NEXT:GOTO 330
      99                                <020>
33022 FOR PX=1 TO LEN(A$):L=0           <174>
33025 IF MID$(A$,PX,1)="@"THEN CB=PX:
      GOTO 33050:IF CC=4 AND CB=0 THE
      N 100                             <240>
33027 NEXT:G9=PQ-6:GOSUB 30700:NEXT:G
      OTO 33099                         <209>
33050 FOR PX=CB TO LEN(A$):IF MID$(A$
      ,PX,1)="@"THEN NEXT               <078>
33060 A$=LEFT$(A$(PQ),CB)+LEFT$(A6$,P
      X-CB-1)                           <158>
33061 X=LEN(A$(PQ))-PX+1:IF X>0 THEN
      A$=A$+RIGHT$(" "+A$(PQ),X)        <010>
33062 A$(PQ)=A$                         <135>
33070 G8=PQ:GOSUB 30000:A$(PQ)=A$:IF
      F=1 THEN 100                      <067>
33072 GOTO 33022                        <208>
33099 CB=A:GOTO 100                     <099>
33200 RETURN                            <237>
34000 REM ***************************
      **                                <229>
34001 REM *
       *                                <011>
34002 REM *  FLOPPY CONTROL
       *                                <036>
34003 REM *
       *                                <013>
34010 REM ***************************
      **                                <239>
34020 INPUT#2,EN,EM$,ET,ES:IF EN=0 TH
      EN RETURN                         <196>
34050 PRINT"{HOME,20DOWN}";EN;"{2SPAC
      E}";EM$                           <009>
34060 PRINT"{2DOWN}TASTE DRUECKEN"      <168>
34070 GOSUB 1                           <224>
34080 IF EN=62 OR XY=999 THEN RETURN    <127>
34090 PRINT"KONTROLLIEREN UND NACH DE
      M BEHEBEN{6SPACE}[F1] DRUECKEN"   <157>
34091 GOSUB 1:IF A$<>"{F1}"THEN 34091   <151>
34092 PRINT#2,"I":GOTO 100              <040>
34100 POKE 915,5:POKE 908,0:POKE 917,
      M8:SYS 50182,A$                   <200>
34110 GOSUB 31:RETURN                   <021>
34500 FOR T=20767+8500 TO 65000         <059>
34510 IF PEEK(T)<>0 THEN NEXT           <017>
34520 X=T                               <032>
34530 PRINT PEEK(X)+PEEK(X+1)*256       <124>
34540 X=X+2                             <085>
34550 IF PEEK(X)=0 THEN X=X+3:PRINT:G
      OTO 34530                         <072>
34560 PRINT CHR$(PEEK(X));:X=X+1:GOTO
       34550                            <243>
40000 PRINT"{HOME,8DOWN}DATUM:":A$="-
      -----":GOSUB 9:DA$=A$             <019>
40010 IF DA$="------"THEN 40000         <118>
40020 D$=LEFT$(DA$,2)+"."+MID$(DA$,3,
      2)+".19"+RIGHT$(DA$,2):RETURN     <104>
43000 REM                               <133>
43040 AM=1                              <007>
43060 RETURN                            <191>
46000 PRINT"{CLR}";PT$;CHR$(14):M8=39
      :CB=1:MB=39:T9%=1                 <058>
46001 PRINT"{DOWN}PARAMETERDISKETTE E
      INLEGEN{DOWN}"                    <065>
46002 IF PEEK(930)<>0 THEN BA$=CHR$(P
      EEK(930))+CHR$(PEEK(931)):FA=PE
      EK(932):GOTO 46070                <039>
46003 PRINT"FLOPPY{2SPACE}ADR.:{2SPAC
      E}8":GOSUB 1:FA=VAL(A$):IF A$=C
      HR$(13)THEN FA=8                  <017>
46004 IF FA<8 OR FA>9 THEN 46003        <253>
46005 T9%=1:PRINT"{2DOWN}NAME{9SPACE}
      ":GOSUB 4:GOSUB 5:B$=A$:T=0:PRI
      NT                                <131>
46006 IF VAL(B$)<>0 THEN Y=VAL(B$):GO
      SUB 4500:GOTO 46010               <093>
46007 GOSUB 4100:IF F<>999 THEN GOSUB
       4500                             <209>
46010 IF F=999 OR EN<>0 THEN PRINT"{D
      OWN}BENUTZER NICHT REGISTRIERT"
      :GOSUB 1:GOTO 46000               <111>
46013 IF PEEK(1022)=200 THEN AM=2       <032>
46050 M8=39                             <030>
46053 IF V$(7)="-"THEN PRINT"KEINE ZU
      GRIFFSBERECHTIGUNG":GOSUB 1:GOT
      O 46000                           <033>
46055 IF T>2 THEN PRINT:PRINT"{WHITE}
      PASSWORT FALSCH":GOSUB 1:GOTO 4
      6000                              <139>
46056 PRINT"{HOME,8DOWN}CODEWORT{BLAC
      K}":GOSUB 4:GOSUB 5:PRINT"{WHIT
      E}":IF A$<>V$(7)THEN T=T+1:GOTO
       46055                            <238>
46058 BA$=V$(8)                         <223>
46060 POKE 930,ASC(LEFT$(BA$,1))        <066>
46061 POKE 931,ASC(RIGHT$(BA$,1))       <101>
46062 POKE 932,FA                       <100>
46070 XY=0:RETURN                       <142>
55555 CLOSE 15:OPEN 15,8,15,"I":CLOSE
       15                               <176>
55556 PRINT"{CLR,4DOWN}ICH SPEICHERE"
      : SAVE"@:PT"+DA$,8                <192>
55557 POKE 2,0                          <111>
55560 PRINT"{HOME,7DOWN}IST GESPEICHE
      RT":END                           <103>
56100 PRINT"{CLR}";PT$                  <097>
56101 PRINT"{2DOWN,SPACE,RVSON}ACHTUN
      G !!!{RVOFF}"                     <201>
56102 PRINT"{DOWN,SPACE}NUR WENN DER
      TEXT SCHON GESPEICHERT"           <165>
56103 PRINT" IST, DANN DIE HYPRA-TEXT
       DISKETTE"                        <246>
56104 PRINT" EINLEGEN, UND [F8] DRUEC
      KEN !!!"                          <044>
56106 GOSUB 1:IF A$<>"{F8}"THEN 100     <194>
56107 PRINT"{CLR}";PT$:PRINT"{2DOWN}I
      CH LADE NUN PROFIPRINT"           <130>
56108 PRINT"{DOWN,2SPACE}BITTE WARTEN
       ..."                             <254>
56109 POKE 2,0:LOAD"C/PROFIPRINT",8     <086>
```


Listing 1. »Hypra-Text«-Hauptprogramm (Schluß)

```
programm : profip v1.0      c400 cbac
--------------------------------------

c400 : 4c 1e c4 4c 05 c9 4c 52  8a
c408 : c9 4c a9 c9 4c 54 ca 4c  c6
c410 : 81 ca 4c aa ca 4c 10 cb  46
c418 : 4c 21 cb 4c 40 cb 20 2b  aa
c420 : c4 20 3b c4 20 71 c4 20  3d
c428 : d8 c8 60 a2 00 8e 94 03  9e
c430 : a0 fe a9 20 99 ff ce 88  a4
c438 : d0 f8 60 ea 20 fd ae 20  e7
c440 : 9e ad 20 a6 b6 aa 8e 93  b4
c448 : 03 a0 00 e8 ca f0 10 b1  90
c450 : 22 99 00 cf c8 c9 0d d0  e9
c458 : f3 20 e5 aa 4c 4c c4 ad  c0
c460 : 8c 03 8d 90 03 a9 00 8d  7c
c468 : 91 03 8d 92 03 8d 97 03  31
c470 : 60 4c 7f c6 ea a5 c6 f0  88
c478 : fc 20 b4 e5 c9 03 f0 f5  d2
c480 : 8d 92 03 ac 9a 03 c0 01  73
c488 : d0 03 20 17 c7 ac 9a 03  17
c490 : c0 02 d0 06 20 17 c7 20  60
c498 : 81 c7 4c 41 c7 c9 9d d0  1b
c4a0 : 14 ae 90 03 d0 08 a9 14  ac
c4a8 : 8d 91 03 4c de c6 ca 8e  b5
c4b0 : 90 03 4c 7f c6 c9 1d d0  96
c4b8 : 17 ae 90 03 e8 ec 95 03  fd
c4c0 : d0 08 a2 00 8e 90 03 4c  4f
c4c8 : 7f c6 8e 90 03 4c 7f c6  7e
c4d0 : c9 aa d0 08 a9 01 8d 99  2f
c4d8 : 03 ad 92 03 c9 a7 d0 08  e4
c4e0 : a9 00 8d 99 03 ad 92 03  0e
c4e8 : c9 13 d0 08 a9 09 8d 91  ac
c4f0 : 03 4c de c6 c9 93 d0 08  36
c4f8 : a9 0f 8d 91 03 4c de c6  5a
c500 : c9 8d d0 08 a9 10 8d 91  39
c508 : 03 4c de c6 c9 85 d0 08  de
c510 : a9 01 8d 91 03 4c de c6  6b
c518 : c9 86 d0 08 a9 03 8d 91  65
c520 : 03 4c de c6 c9 87 d0 08  06
c528 : a9 05 8d 91 03 4c de c6  85
c530 : c9 88 d0 08 a9 07 8d 91  9f
c538 : 03 4c de c6 c9 89 d0 08  2e
c540 : a9 02 8d 91 03 4c de c6  1b
c548 : c9 8a d0 08 a9 04 8d 91  a0
c550 : 03 4c de c6 c9 8b d0 08  56
c558 : a9 06 8d 91 03 4c de c6  35
c560 : c9 8c d0 08 a9 08 8d 91  d9
c568 : 03 4c de c6 ae 90 03 ec  77
c570 : 96 03 d0 30 a9 15 8d 01  3d
c578 : d4 a9 50 8d 00 d4 a9 0f  52
c580 : 8d 18 d4 a9 19 8d 05 d4  3f
c588 : a9 14 8d 06 d4 a9 11 8d  59
c590 : 04 d4 a0 50 a2 ff ca d0  27
c598 : fd 88 d0 f8 a9 00 8d 18  2d
c5a0 : d4 8d 04 d4 ae 90 03 ad  ad
c5a8 : 92 03 c9 0d d0 03 4c de  e4
c5b0 : c6 c9 14 d0 22 ae 95 03  6e
c5b8 : e8 8e 93 03 ae 90 03 f0  8a
c5c0 : 13 bd 00 cf 9d ff ce e8  93
c5c8 : ec 93 03 d0 f2 ae 90 03  46
c5d0 : ca 8e 90 03 4c 7f c6 c9  d5
c5d8 : 94 d0 1a ae 90 03 ac 95  30
c5e0 : 03 b9 ff ce 99 00 cf 88  83
c5e8 : cc 90 03 d0 f4 a9 20 9d  30
c5f0 : 00 cf 4c d4 c5 c9 91 d0  18
c5f8 : 27 ae 90 03 e8 bd 00 cf  17
c600 : c9 20 f0 12 c9 00 ec 95  d3
c608 : 03 d0 f1 ae 95 03 ca ca  f8
c610 : 8e 90 03 4c 7f c6 e8 4c  9b
c618 : 1a c6 8e 90 03 4c 7f c6  69
c620 : c9 11 d0 23 ae 90 03 e0  48
c628 : 00 d0 03 4c 74 c4 ca ca  09
c630 : bd 00 cf c9 20 f0 0c e0  96
c638 : 00 30 d0 d0 f2 8e 90 03  8a
c640 : 4c 7f c6 e8 4c 3d c6 ad  40
c648 : 92 03 c9 20 d0 0b ae 90  13
c650 : 03 ec 96 03 30 03 4c de  d9
c658 : c6 ae 90 03 9d 00 cf e8  e5
c660 : 8e 90 03 ac 99 03 c0 02  45
c668 : d0 09 a9 a0 9d 00 cf e8  26
c670 : 8e 90 03 ac 99 03 c0 01  53
c678 : d0 05 a0 02 8c 99 03 ae  32
c680 : 98 03 ac 9c 03 20 0c e5  86
c688 : a2 00 ac 95 03 bd 00 cf  c6
c690 : 20 47 ab e8 88 f0 06 20  c4
c698 : e5 aa 4c 8d c6 ad 90 03  b9
c6a0 : 18 6d 9c 03 a8 ae 98 03  5f
c6a8 : 20 0c e5 a9 12 20 47 ab  13
c6b0 : ac 90 03 b9 00 cf 20 47  2a
c6b8 : ab a9 92 20 47 ab ae 90  8e
c6c0 : 03 ec 95 03 10 08 ae 90  1c
c6c8 : 03 30 0b 4c 74 c4 a9 0b  5a
c6d0 : 8d 91 03 4c de c6 a9 0c  53
c6d8 : 8d 91 03 4c de c6 ae 98  88
c6e0 : 03 ac 9c 03 20 0c e5 a2  00
c6e8 : 00 ac 95 03 bd 00 cf 20  5f
c6f0 : 47 ab e8 88 f0 06 20 e5  e3
c6f8 : aa 4c ec c6 ea ad 90 03  41
c700 : 8d 9d 03 bd 00 cf c9 20  ba
c708 : d0 08 ca f0 03 4c fd c6  c5
c710 : a2 00 e8 8e 90 03 60 ea  37
c718 : c9 59 d0 06 a9 5a 8d 92  4b
c720 : 03 60 c9 d9 d0 06 a9 da  9a
c728 : 8d 92 03 60 c9 5a d0 06  8a
c730 : a9 59 8d 92 03 60 c9 da  4c
c738 : d0 06 a9 d9 8d 92 03 60  eb
```

Listing 2. Maschinenroutine zu »Hypra-Text«. Bei der Eingabe bitte den MSE verwenden.

```
c740 : 60 ea c9 81 f0 2b c9 95  73
c748 : f0 27 c9 96 f0 23 c9 97  8f
c750 : f0 1f c9 98 f0 1b c9 99  97
c758 : f0 17 c9 9a f0 13 c9 9b  9f
c760 : f0 0f c9 12 f0 0b c9 92  40
c768 : f0 07 c9 22 f0 0b 4c 9d  66
c770 : c4 a9 00 8d 92 03 4c 74  16
c778 : c4 a9 b6 8d 92 03 4c 9d  1e
c780 : c4 ad 92 03 4c 2e c8 c9  0d
c788 : 3a d0 06 a9 b1 8d 92 03  b9
c790 : 60 c9 5b d0 06 a9 b2 8d  59
c798 : 92 03 60 c9 3b d0 06 a9  a3
c7a0 : a3 8d 92 03 60 c9 5d d0  7a
c7a8 : 06 a9 b7 8d 92 03 60 c9  79
c7b0 : 40 d0 06 a9 b8 8d 92 03  57
c7b8 : 60 c9 ba d0 06 a9 a2 8d  19
c7c0 : 92 03 60 c9 2b d0 06 a9  ca
c7c8 : a6 8d 92 03 60 c9 3c d0  21
c7d0 : 06 a9 3b 8d 92 03 60 c9  82
c7d8 : 3e d0 06 a9 3a 8d 92 03  95
c7e0 : 60 c9 23 d0 06 a9 af 8d  8f
c7e8 : 92 03 60 c9 3d d0 06 a9  13
c7f0 : 23 8d 92 03 60 c9 27 d0  71
c7f8 : 06 a9 2f 8d 92 03 60 c9  a7
c800 : db d0 06 a9 3f 8d 92 03  ab
c808 : 60 c9 2f d0 06 a9 2d 8d  b0
c810 : 92 03 60 c9 2a d0 06 a9  0a
c818 : 2b 8d 92 03 60 c9 c0 d0  08
c820 : 06 a9 2a 8d 92 03 60 60  bb
c828 : 60 60 60 60 60 60 ad 8e  b9
c830 : 02 c9 01 d0 03 4c be c8  90
c838 : c9 02 f0 0d c9 04 d0 03  e6
c840 : 4c 85 c8 ad 92 03 4c 87  b8
c848 : c7 ad 92 03 c9 3c d0 06  b9
c850 : a9 3c 8d 92 03 60 c9 3e  a4
c858 : d0 06 a9 3e 8d 92 03 60  98
c860 : c9 5b d0 06 a9 5b 8d 92  9d
c868 : 03 60 c9 5d d0 06 a9 5d  58
c870 : 8d 92 03 60 c9 3d d0 06  e9
c878 : a9 40 8d 92 03 60 ea ad  31
c880 : 92 03 4c 87 c7 ad 92 03  d2
c888 : c9 85 d0 08 a5 01 8d 9b  19
c890 : 03 4c 87 c7 c9 86 d0 08  b8
c898 : a5 02 8d 9b 03 4c 87 c7  55
c8a0 : c9 87 d0 08 a5 03 8d 9b  42
c8a8 : 03 4c 87 c7 c9 88 d0 08  e0
c8b0 : a5 04 8d 9b 03 4c 87 c7  6e
c8b8 : a9 20 8d 92 03 60 ad 92  36
c8c0 : 03 c9 30 d0 06 a9 3d 8d  8c
c8c8 : 92 03 60 c9 3d d0 06 a9  f3
c8d0 : 27 8d 92 03 60 4c 87 c7  d9
c8d8 : 20 fd ae 20 8b b0 85 49  8d
c8e0 : 84 4a 20 a3 b6 ad 90 03  27
c8e8 : 20 75 b4 a0 02 b9 61 00  77
c8f0 : 91 49 88 10 f8 c8 a2 00  aa
c8f8 : bd 00 cf 91 62 e8 c8 ec  46
c900 : 90 03 d0 f4 60 20 2b c4  22
c908 : 20 3b c4 a9 00 8d 97 03  fd
c910 : ae 93 03 e8 20 fd c6 8e  90
c918 : 93 03 bd 00 cf c9 bb f0  b8
c920 : 2a c9 a5 f0 26 c9 b4 f0  1c
c928 : 22 c9 b5 f0 1e c9 a1 f0  53
c930 : 1a c9 aa f0 16 c9 a7 f0  28
c938 : 12 c9 bc f0 0e c9 be f0  88
c940 : 0a c9 ac f0 06 ca 30 09  02
c948 : 4c 1a c9 ee 97 03 4c 45  3f
c950 : c9 60 20 2b c4 ae 95 03  d5
c958 : e8 8e 95 03 ad 93 03 ae  2e
c960 : 93 03 ac 93 03 20 1e e1  80
c968 : a2 00 e8 20 12 e1 c9 0d  ba
c970 : f0 16 ac 8c 03 c0 ff d0  00
c978 : 04 c9 20 f0 26 9d ff ce  74
c980 : ec 95 03 f0 03 4c 6a c9  e6
c988 : 8d 91 03 8e 90 03 20 fd  0e
c990 : c6 8e 90 03 a9 00 85 99  06
c998 : ae 95 03 ca 8e 95 03 20  0d
c9a0 : d8 c8 60 ca f0 c2 4c 88  b5
c9a8 : c9 20 05 c9 a9 01 85 9b  ec
c9b0 : a9 00 8d 91 03 ad 97 03  f1
c9b8 : 8d 8c 03 ad 93 03 18 ed  8f
c9c0 : 97 03 8d 97 03 ee 97 03  3b
c9c8 : ad 97 03 cd 96 03 30 46  8a
c9d0 : cd 95 03 f0 41 ae 91 03  1d
c9d8 : 10 0f ae 93 03 ca 8e 91  72
c9e0 : 03 e6 9b a5 9b c9 1e f0  54
c9e8 : 2d bd 00 cf c9 20 f0 20  8f
c9f0 : ca 8e 91 03 4c c8 c9 18  28
c9f8 : ad 91 03 e5 9b 8d 91 03  5e
ca00 : cd 95 03 10 03 4c c8 c9  a4
ca08 : a2 00 8e 91 03 4c c8 c9  c9
ca10 : 20 1e ca 4c f7 c9 a2 f0  b6
ca18 : 20 fd c6 4c d8 c8 ae 91  24
ca20 : 03 ec 95 03 30 01 60 ae  49
ca28 : 91 03 10 01 60 ad 95 03  2f
ca30 : 6d 8c 03 a8 b9 ff ce 99  c3
ca38 : 00 cf 88 cc 91 03 10 f4  37
```

```
0 DATA 169,240,32,71,3,169,160,32,71,
  3,96,133,89,120,169,51,133,1,169,0  <184.
1 DATA 169,0,133,88,133,90,133,95,169
  ,224,133,91,169,208,133,96,32,191,1
  63                                  <004>
2 DATA 169,160,169,55,133,1,88,96     <076>
1000 FOR T=0 TO 46:READ X:POKE T+828,
     X:NEXT                           <220>
1010 SYS 828                          <093>
1060 POKE 53272,8:POKE 56576,PEEK(565
     76)AND 252:POKE 648,192:PRINT"{C
     LR}"                             <117>
1100 FOR T=1 TO 128:READ A$,X:POKE 10
     26+T,X:IF A$<>"END"THEN NEXT     <071>
1110 PRINT 1027+T                     <012>
2000 GOTO 5000                        <194>
3000 Z1=36864+T*8+X+20472+7           <248>
3040 POKE Z1,X(X)                     <146>
3050 POKE Z1+256*8,X(X)               <194>
3051 POKE Z1+128*8,255-X(X)           <013>
3052 POKE Z1+384*8,255-X(X)           <141>
3060 Z1=Z1-5*16↑3                     <230>
3070 POKE Z1,X(X)                     <176>
3080 POKE Z1+256*8,X(X)               <226>
3081 POKE Z1+128*8,255-X(X)           <045>
3082 POKE Z1+384*8,255-X(X)           <173>
3999 RETURN                           <247>
4000 REM                              <252>
4010 FOR X=1 TO 8                     <038>
4020 GOSUB 3000:NEXT                  <085>
4030 PRINT"{3DOWN}":GOTO 5030         <182>
4900 :                                <050>
5000 XX=1200:POKE 1024,8:POKE 1025,15
     :POKE 1026,14                    <089>
5030 READ A$:A$(1)=A$:IF A$="END"THEN
      10000                           <131>
5040 FOR T=1 TO 8:READ X(T):IF X(1)=9
     99 THEN 5042                     <066>
5041 NEXT                             <225>
5042 FOR T=1 TO 8:READ Y(T):NEXT      <062>
5050 GOSUB 6500                       <044>
5055 IF X(1)=999 THEN 5030            <041>
5060 GOTO 6000:GOTO 5030              <066>
6000 PRINT"{CLR}";A$(1):PRINT PEEK(49
     152)                             <118>
6010 T=PEEK(49152):GOTO 4000          <157>
6500 REM ** DRUCKERCODE **            <091>
6510 AS=ASC(A$):AS=AS-160             <062>
6520 AS=(AS*8)+1200:FOR T=1 TO 8:D(T)
     =Y(T):NEXT                       <088>
6540 FOR T=1 TO 8:POKE AS+T,D(T):NEXT <210>
6550 RETURN                           <002>
7000 :                                <118>
8499 DATA "END",0                     <095>
9999 DATA"END",0,0,0,0,0,0,0,0,0,0,
     0,0,0,0,0,0,,0,0,0,0,,0,0,0,0,0,
     0,0                              <106>
10000 PRINT"{CLR}":GOTO 12000         <125>
12000 GOSUB 40000                     <064>
13000 PRINT"{CLR,4DOWN}DAS PROGRAMM W
      IRD GESPEICHERT":POKE 46,38     <025>
13001 PRINT"{DOWN,2SPACE}- BITTE WART
      EN -{2SPACE}"                   <158>
13003 SAVE"@:LOADER DIN 2.1",8        <028>
13004 VERIFY"LOADER DIN 2.1",8        <161>
13005 PRINT"DAS PROGRAMM IST GESPEICH
      ERT":END                        <170>
20000 X= 160 :L= 8500                 <232>
20001 IF X>191 THEN 30000             <043>
20010 PRINT"{CLR,2DOWN}":PRINT"20000
      X=";X+1;":L=";L+2               <105>
20020 PRINT L;"DATA"+CHR$(34)+CHR$(X)
      +CHR$(34);",999,";X;",0,0,0,0,0
      ,0,0"                           <114>
20030 PRINT"GOTO20000{HOME}"          <042>
20040 POKE 198,5:FOR T=631 TO 631+5:P
      OKE T,13:NEXT:END               <172>
30000 X= 1:L= 8000                    <175>
30001 IF X>127 THEN PRINT"{CLR}FERTIG
      ":END                           <115>
30010 PRINT"{CLR,2DOWN}":PRINT"30000
      X=";X+1;":L=";L+2               <207>
30019 IF X=34 THEN PRINT L;"DATA"+CHR
      $(34)+"''"+CHR$(34);",";X:GOTO
      30030                           <000>
30020 IF X>31 THEN PRINT L;"DATA"+CHR
      $(34)+CHR$(X)+CHR$(34);",";X    <197>
30021 IF X<32 THEN PRINT L;"DATA"+CHR
      $(34)+"."+CHR$(34);",";X        <134>
30030 PRINT"GOTO30000{HOME}"          <137>
30040 POKE 198,5:FOR T=631 TO 631+5:P
      OKE T,13:NEXT:END               <010>
40000 REM *** SAVE FILE ***           <219>
40001 OPEN 2,8,1,"@:ZEICHEN DDIN"     <181>
40010 PRINT#2,CHR$(0);CHR$(4);        <135>
40020 FOR T=4*16↑2 TO 2048:PRINT#2,CH
      R$(PEEK(T));:NEXT               <106>
40030 CLOSE 2                         <170>
40040 OPEN 2,8,1,"@:ZEICHEN DIN"      <176>
40050 PRINT#2,CHR$(0);CHR$(14*16);    <073>
40060 FOR T=9*16↑3 TO 9*16↑3+4096     <155>
40070 PRINT#2,CHR$(PEEK(T));:NEXT     <046>
40080 CLOSE 2:RETURN                  <080>
```

© 64'er

Listing 3. Mit dem Programm »ZLOADER« definieren Sie Bildschirm- und Druckersteuerzeichen. Beachten Sie bitte die Eingabehinweise auf Seite 54.


```
ca40 : a9 20 9d 00 cf ac 93 03  17
ca48 : c8 8c 93 03 ac 97 03 c8  c1
ca50 : 8c 97 03 60 a5 a4 8d 67  f9
ca58 : ca a5 a5 8d 68 ca 78 a9  22
ca60 : 35 85 01 ac 95 03 b9 33  ec
ca68 : 33 99 00 cf c0 00 f0 04  3a
ca70 : 88 4c 66 ca a9 37 85 01  7e
ca78 : ad 95 03 8d 90 03 4c d8  66
ca80 : c8 20 2b c4 20 3b c4 78  9c
ca88 : a9 35 85 01 a5 a4 8d 9d  3e
ca90 : ca a5 a5 8d 9e ca ac 95  66
ca98 : 03 b9 00 cf 99 00 50 88  5e
caa0 : 30 03 4c 99 ca a9 37 85  7a
caa8 : 01 60 78 a9 ff 8d 03 dd  61
cab0 : ad 02 dd 09 04 8d 02 dd  67
cab8 : ad 00 d0 09 04 8d 00 d0  09
cac0 : a9 10 8d 0d dd ad 0d dd  b2
cac8 : a9 e3 8d 26 03 a9 ca 8d  4f
cad0 : 27 03 a9 1b 20 f7 ca a9  87
cad8 : 43 20 f7 ca a9 48 20 f7  d0
cae0 : ca 58 60 48 a5 9a c9 04  56
cae8 : f0 03 4c cd f1 68 c9 0d  ca
caf0 : d0 05 20 f7 ca a9 0a 8d  87
caf8 : 01 dd ad 00 dd 29 fb 8d  85
cb00 : 00 dd 09 04 8d 00 dd ad  5d
cb08 : 0d dd 29 10 f0 f9 18 60  50
cb10 : 20 2b c4 20 3b c4 ac 93  af
cb18 : 03 88 8c 90 03 20 d8 c8  bb
cb20 : 60 20 2b c4 20 3b c4 a0  24
cb28 : 00 b9 01 cf 99 00 cf c8  a9
cb30 : cc 93 03 d0 f4 ac 93 03  aa
cb38 : 88 8c 90 03 20 d8 c8 60  38
cb40 : a9 60 85 a5 a9 00 85 a4  29
cb48 : a0 ff a9 20 91 a4 88 d0  58
cb50 : f9 e6 a5 a5 a5 c9 c0 d0  28
cb58 : f1 60 a5 02 8d 20 d0 ad  9b
cb60 : 92 03 c9 ff d0 02 a9 7e  a7
cb68 : c9 e0 30 03 38 e9 40 c9  75
cb70 : c0 30 03 38 e9 60 c9 a0  1a
cb78 : 30 26 38 e9 a0 0a 0a 0a  9d
cb80 : 48 a9 a2 85 9e a9 04 85  48
cb88 : 9f 90 06 a9 00 65 9e 85  d7
cb90 : 9e 68 a8 b1 9e f0 08 48  e5
cb98 : 20 ee ca c8 4c 93 cb 60  4c
cba0 : a8 a9 01 8d 20 d0 b9 02  82
cba8 : 04 4c ee ca ff ff ff ff  e7
```

Listing 2. (Schluß)

```
8500 DATA"{SHIFT-SPACE}",0,0,0,24,0,0
     ,0,0, 32 ,0,0,0,0,0,0,0          <156>
8502 DATA"⌈",0,112,34,38,34,112,0,0,
     27 ,53,0,0,0,0,0,0               <226>
8504 DATA"⌈",102,0,102,102,102,102,60
     ,0,93,0,0,0,0,0,0,0              <021>
8506 DATA"⌈",102,0,60,6,62,102,62,0,1
     23,0 ,0,0,0,0,0,0                <112>
8508 DATA"⌈",999, 241 ,0,0,0,0,0,0,0  <175>
8510 DATA"⌈",0,12,74 ,106,74,12,0,0,
     27 ,69,0,0,0,0,0,0               <214>
8512 DATA"⌈",60,54,60,54,60,48,96,0,1
     26,000,000,000,000,000,0,0       <034>
8514 DATA"⌈",0,224,178,230,178,224,0,
     0, 27 ,87,48,0,0,0,0,0           <222>
8516 DATA"⌈",999,168,0,0,0,0,0,0,0    <240>
8518 DATA"£",60,66,153,161,161,153,66
     ,60,203 ,0,0,0,0,0,0,0           <116>
8520 DATA"⌈",0,28,150,220,150,28,0,0,
      27 ,87,49,0,0,0,0,0             <014>
8522 DATA"⌈",999, 244 ,0,0,0,0,0,0,0  <141>
8524 DATA"⌈",0,0,124,56,16,0,0,0, 32
     ,0,0,0,0,0,0,0                   <025>
8526 DATA"⌈",999, 246 ,0,0,0,0,0,0,0  <241>
8528 DATA"⌈",999, 242 ,0,0,0,0,0,0,0  <131>
8530 DATA"⌈",12,26,24,44,24,88,48,0,6
     4,0,000,000,000,000,0,0          <244>
8532 DATA"⌈",999, 240 ,0,0,0,0,0,0,0  <104>
8534 DATA"⌈",102,0,60,102,102,102,60,
     0,124,000,000,000,000,000,0,0    <245>
8536 DATA"⌈",102,60,102,102,102,102,6
     0,0,92,000,000,000,000,000,0,0   <010>
8538 DATA"⌈",999, 249,0,0,0,0,0,0,0   <191>
8540 DATA"⌈",0,96 ,82 ,86 ,82 ,96,0,0
     , 27  ,70,0,0,0,0,0,0            <024>
8542 DATA"⌈",0,28,136,200,136,28,0,0,
      27 ,52,0,0,0,0,0,0              <131>
8544 DATA"⌈",102,102,102,0,0,0,0,0,34
      ,0,0,0,0,0,0,0                  <131>
8546 DATA"⌈",102,24,60,102,126,102,10
     2,0,91,000,000,000,000,000,0,0   <024>
8548 DATA"⌈",102,0,102,102,102,102,59
     ,0,125,000,000,000,000,000,0,0   <015>
8550 DATA"⌈",999,185,0,0,0,0,0,0,0    <226>
8552 DATA"⌈",999, 186 ,0,0,0,0,0,0,0  <020>
8554 DATA"⌈",0,16,48,112,48,16,0,0, 3
     2  ,0,0,0,0,0,0,0                <124>
8556 DATA"⌈",0,18,146,210,146,12,0,0,
      27 ,45,49,0,0,0,0,0             <228>
8558 DATA"⌈",999, 189 ,0,0,0,0,0,0,0  <171>
8560 DATA"⌈",0,144,146,150,146,96,0,0
     ,27 ,45,48,0,0,0,0,0             <204>
8562 DATA"⌈",999, 191 ,0,0,0,0,0,0,0  <222>
8564 DATA"⌈",999, 241 ,0,0,0,0,0,0,0  <169>
```

© 64'er

Listing 4. Änderung für die Zeilen 8500 bis 9000



Pac-Man unter der Lupe

Eine ganz besondere Pac-Man-Version bieten wir Ihnen mit diesem Listing. Ihnen steht ein riesiges Labyrinth zur Verfügung. Um den Pac-Man hier noch zu erkennen, mußte er unter die Lupe genommen werden.

Nachdem ich des Spielens müde auf einem Apple ein Pac-Man-Spiel sah, das es in dieser Form auf dem C 64 noch nicht gab, erwachte der Programmierer in mir. Ich wollte dieses Spiel für den C 64 entwickeln.

Gegenüber dem Apple hat der C 64 Sprites zur Verfügung, die man sogar vergrößern kann. Also mußte man nur noch eine Bildschirm/Sprite-Verschieberoutine schreiben. Als diese Routine funktionierte, mußte das Spielfeld erstellt werden. Dafür wurde ein spezieller Line-Befehl geschrieben, der mit einem Minimum an Parametern über Spiegelung das Spielfeld erzeugt. Die Joystickabfrage wurde an Port 2 gelegt, da dies die meisten Spiele tun.

Der Verfolger-Algorithmus war anfänglich darauf ausgelegt »intelligent« zu sein. Aber bei dieser Steuerung wurde man spätestens nach dem Fressen von fünf Punkten eingeholt. So wurde ein einfacherer Algorithmus gewählt, der nur abläuft, wenn man sich selbst bewegt.

Noch ein wichtiger Hinweis: Nach dem ersten RUN wird das Programm »decompressed«, das heißt die Maschinenroutinen und die Grafik werden absolut im Speicher abgelegt. Jetzt kann man mit LIST ein kleines Basic-Programm einsehen, dessen Aufgabe es ist, die Verfolger und die Farben zu setzen. In Zeile 1 ändert man die Bildschirmfarbe. Hier muß die 110 ersetzt werden (Werte siehe Handbuch). In Zeile 2 wird durch Ändern der 110 die Score-Farbe gewählt. In Zeile 3 wird die Lupenfarbe eingestellt. Durch Ersetzen der 4 in Zeile 4 (Werte von 1 bis 8) kann die Anzahl der Verfolger geändert werden. Hat man nun die Farben eingestellt (oder auch nicht) wird das Spiel ein zweites Mal mit RUN gestartet.

Steuern können Sie mit dem Joystick in Port 2. Wird man gefressen, startet man ein neues Spiel mit dem Feuerknopf.

Eine kurze Bemerkung für Maschinensprache-Freaks: Wer Lust und Laune zu einem besseren Ton hat, der kann einfach Musik in den IRQ legen. Das Programm läuft in einer einzigen Jobschleife ab.

(Thomas Kaiser/rg)

```
PROGRAMM : GHOST                 0801 1DDA
-------------------------------------------

0801 : 25 08 DA 07 9E 32 30 38   6F
0809 : 37 20 54 48 4F 4D 41 53   7A
0811 : 20 4B 41 49 53 45 52 20   39
0819 : 31 39 38 35 20 20 20 20   5F
0821 : 20 20 20 00 00 00 A2 00   E4
0829 : BD 38 08 9D 00 01 E8 D0   05
0831 : F7 A2 FF 9A 4C 00 01 78   86
0839 : A9 34 85 01 A0 00 A9 DA   E4
0841 : 85 AE A9 1D 85 AF A9 00   A8
0849 : 85 AC 85 AD A5 AC D0 02   42
0851 : C6 AD C6 AC A5 AE D0 02   4C
0859 : C6 AF C6 AE B1 AE 91 AC   AE
0861 : A5 AE C9 BE D0 E6 A5 AF   E2
0869 : C9 08 D0 E0 A9 01 85 AE   9D
0871 : A9 08 85 AF B1 AC C9 BF   9D
0879 : D0 12 20 78 01 B1 AC AA   0F
0881 : A9 00 91 AE 20 7F 01 CA   FC
0889 : D0 F8 F0 16 C9 CF D0 0D   4D
0891 : 20 78 01 B1 AC AA 20 78   F5
0899 : 01 B1 AC D0 E5 91 AE 20   9E
08A1 : 7F 01 20 78 01 D0 CD A9   D9
08A9 : 37 85 01 58 4C 7A A8 E6   F7
08B1 : AC D0 02 E6 AD 60 E6 AE   FA
08B9 : D0 02 E6 AF 60 0E 08 01   D2
08C1 : 00 97 38 34 30 2C 31 31   AD
08C9 : 30 00 1B 08 02 00 97 38   B0
08D1 : 34 31 2C 31 31 30 00 28   B4
08D9 : 08 03 00 97 38 34 32 2C   9C
08E1 : 31 31 30 00 33 08 04 00   3B
08E9 : 97 38 34 33 2C 34 00 3E   F1
08F1 : 08 0A 00 9E 35 32 30 33   DE
08F9 : 30 00 4D 08 14 00 97 32   82
0901 : 35 35 39 32 2C 31 34 34   EB
0909 : 00 5C 08 1E 00 97 32 35   ED
0911 : 35 39 33 2C 31 34 35 00   BF
0919 : 67 08 28 00 9E 35 32 CF   8A
0921 : 03 30 00 6D 08 32 00 8A   11
0929 : BF 03 CF 00 AA CF 00 AA   DC
0931 : CF 00 AA CF 00 AA CF 00   39
0939 : AA CF 00 AA CF 00 AA CF   67
0941 : 00 AA CF 00 AA CF 00 AA   09
0949 : CF 00 AA CF 00 AA CF 00   51
0951 : AA CF 00 AA CF 00 AA CF   7F
0959 : 00 AA CF 00 AA CF 00 AA   21
0961 : CF 00 AA CF 00 AA CF 00   69
0969 : AA CF 00 AA CF 00 AA CF   97
0971 : 00 AA CF 00 AA CF 00 AA   39
0979 : CF 00 AA CF 00 AA CF 00   81
0981 : AA CF 00 AA CF 00 AA CF   AF
0989 : 00 AA CF 00 AA CF 00 AA   51
0991 : CF 00 AA CF 00 AA CF 00   99
0999 : AA CF 00 AA CF 00 AA CF   C7
09A1 : 00 AA CF 00 AA CF 00 AA   69
09A9 : CF 00 AA CF 00 AA CF 00   B1
09B1 : AA CF 00 AA CF 00 AA CF   DF
09B9 : 00 AA CF 00 AA CF 00 AA   81
09C1 : CF 00 AA CF 00 AA CF 00   C9
09C9 : AA CF 00 AA CF 00 AA CF   F7
09D1 : 00 AA CF 00 AA CF 00 AA   99
09D9 : CF 00 AA CF 00 AA CF 00   E1
09E1 : AA CF 00 AA CF 00 AA CF   0F
09E9 : 00 AA CF 00 AA CF 00 AA   B1
09F1 : CF 00 AA CF 00 AA CF 00   F9
09F9 : AA CF 00 AA CF 00 AA CF   27
0A01 : 00 AA CF 00 AA CF 00 AA   C9
0A09 : CF 00 AA CF 00 AA CF 00   11
0A11 : AA CF 00 AA CF 00 AA CF   3F
0A19 : 00 AA CF 00 AA CF 00 AA   E1
0A21 : CF 00 AA CF 00 AA CF 00   29
0A29 : AA CF 00 AA CF 00 AA CF   57
0A31 : 00 AA CF 00 AA CF 00 AA   F9
0A39 : CF 00 AA CF 00 AA CF 91   65
0A41 : AA 00 00 A0 BF 04 64 04   B5
0A49 : 04 08 CF 04 04 0C 04 0C   8F
0A51 : 08 0C 04 10 04 20 04 20   F4
0A59 : 08 20 08 10 08 1C 08 1C   2F
0A61 : 0C 1C 0C 1C 0C 2C 0C 2C   AD
0A69 : 10 2C 00 24 08 24 08 24   1E
0A71 : 08 30 04 28 04 34 08 08   AA
0A79 : 18 08 0C 04 0C 18 10 04   E3
0A81 : 1C 04 1C 08 1C 0C 10 0C   22
0A89 : 1C 0C 20 04 38 04 20 08   68
0A91 : 20 CF 03 10 20 CF 04 10   0C
0A99 : 28 10 28 20 28 24 08 28   EC
0AA1 : 08 20 0C 28 0C 28 0C 28   44
0AA9 : 10 14 14 20 CF 04 14 24   82
0AB1 : 14 24 18 24 24 10 24 18   E6
0AB9 : 1C 18 24 CF 04 18 1C 18   86
0AC1 : 1C 28 1C 18 20 1C 20 1C   97
0AC9 : 20 1C 24 1C 24 2C 24 2C   11
0AD1 : 1C 2C 24 20 20 28 20 2C   2D
0AD9 : 08 2C 14 28 14 2C 14 28   45
0AE1 : 18 30 18 30 08 34 08 38   D0
0AE9 : 08 38 18 30 0C 48 0C 30   AD
0AF1 : 0C 30 10 34 10 34 14 30   F3
0AF9 : 14 34 14 28 18 30 18 30   F5
0B01 : 18 30 30 04 34 0C 34 0C   4B
0B09 : 30 0C 34 14 28 14 30 10   D3
0B11 : 30 18 30 10 30 10 38 0C   D8
0B19 : 38 10 38 04 38 04 4C 08   CD
0B21 : 38 08 40 08 3C 14 3C 14   EC
0B29 : 34 14 3C 14 34 1C 34 18   1E
0B31 : 2C 24 2C 24 28 24 2C CF   F3
0B39 : 03 28 30 20 30 28 30 2C   BE
0B41 : 28 2C 38 1C 30 1C 48 20   56
0B49 : 30 20 3C 24 34 28 34 24   BB
0B51 : 34 24 38 30 24 38 24 34   A8
0B59 : 18 34 20 0C 40 0C 44 08   9B
0B61 : 44 0C 44 18 38 18 40 10   25
0B69 : 40 18 40 10 40 10 58 08   BD
0B71 : 48 10 48 04 4C 0C 4C 0C   C2
0B79 : 4C 0C 50 04 50 04 5C 04   FF
0B81 : 5C 14 5C 08 50 08 54 08   A6
0B89 : 54 10 54 08 58 0C 58 14   6B
0B91 : 44 14 5C 14 44 18 44 18   BF
0B99 : 48 18 4C 18 4C 20 4C 18   2B
0BA1 : 50 1C 50 18 54 18 58 18   AE
0BA9 : 54 20 54 04 60 18 60 1C   23
0BB1 : 58 1C 5C 18 5C 1C 5C 18   7A
0BB9 : 5C 18 60 20 44 20 60 20   45
0BC1 : 3C 28 3C 20 40 24 40 20   8B
0BC9 : 44 28 44 24 48 24 5C 28   1E
0BD1 : 48 28 48 24 60 2C 60 28   FD
0BD9 : 38 28 48 28 CF 03 3C 2C   9B
0BE1 : 40 2C 44 2C 44 30 44 28   F5
0BE9 : 48 38 48 28 4C 2C 4C 2C   14
0BF1 : 4C 2C 54 28 50 28 58 28   65
0BF9 : 58 2C 58 1C 60 20 60 34   F2
0C01 : 28 34 34 30 CF 03 34 2C   95
0C09 : CF 04 38 28 38 38 30 CF   93
0C11 : 03 40 34 40 34 48 30 48   20
0C19 : 30 50 2C 54 34 54 34 4C   56
0C21 : 34 54 30 58 38 58 30 5C   56
0C29 : 30 60 30 60 3C 60 34 5C   F2
0C31 : 38 5C 38 50 38 58 34 4C   5F
0C39 : 3C 4C 3C 48 3C 4C 38 CF   5A
0C41 : 03 44 3C 2C CF 03 3C 2C   59
0C49 : 40 2C 2C 5C 2C 60 24 5C   45
0C51 : 28 5C 34 20 48 20 3C 20   6F
0C59 : 3C 28 38 28 40 28 3C 2C   4B
0C61 : 40 2C 40 30 CF 03 40 48   74
0C69 : 40 50 38 50 40 50 3C 54   0A
0C71 : 3C 60 3C 54 44 54 44 28   BF
0C79 : 44 44 40 24 4C 24 3C 10   6B
0C81 : 3C 1C 38 1C 3C 1C 3C 04   FB
0C89 : 3C 08 3C 08 60 08 40 10   41
0C91 : 48 10 40 10 40 20 44 14   32
0C99 : 44 1C 40 04 64 04 4C 08   24
0CA1 : 4C 14 44 14 4C 14 48 18   42
0CA9 : 50 18 44 1C 54 1C 4C 20   32
0CB1 : 4C 24 50 20 50 28 48 28   DF
0CB9 : 50 28 48 28 48 30 4C 2C   C4
0CC1 : 4C 38 48 34 48 48 40 CF   29
0CC9 : 03 48 44 4C 44 54 40 58   24
0CD1 : 48 58 48 50 48 58 40 5C   62
0CD9 : 4C 5C 40 5C 40 60 44 4C   A0
```

```
0CE1 : 48 4C 4C 40 4C 5C 48 3C   AC
0CE9 : 54 3C 44 60 50 60 50 0C   DA
0CF1 : 58 0C 50 10 60 10 50 10   4D
0CF9 : 50 18 54 14 54 18 54 14   6C
0D01 : 58 14 5C 0C 5C 20 58 18   54
0D09 : 58 40 54 20 54 2C 4C 2C   CB
0D11 : 54 2C 50 30 50 34 50 34   E6
0D19 : 54 34 54 34 54 3C 4C 38   EC
0D21 : 50 38 50 40 60 40 50 40   73
0D29 : 50 4C 50 4C 54 CF 04 50   B1
0D31 : 54 50 54 5C 54 50 CF 03   5B
0D39 : 58 50 5C 50 60 50 5C 54   7D
0D41 : 5C 50 CF 03 58 54 44 54   FB
0D49 : 48 54 44 64 44 58 48 58   32
0D51 : 50 58 48 60 48 5C 4C 5C   3D
0D59 : 58 58 5C 58 60 54 60 64   F2
0D61 : 60 5C 5C 64 5C 5C 28 5C   95
0D69 : 3C 5C 3C 64 4C 5C 4C 64   11
0D71 : 4C 58 24 60 38 5C 20 60   A6
0D79 : 20 60 14 60 1C 60 1C 64   D8
0D81 : 2C 64 08 64 0C 60 0C 64   2B
0D89 : 0C 64 10 64 18 68 00 68   EE
0D91 : 10 6C 04 6C 14 68 14 6C   14
0D99 : 14 68 18 70 18 68 1C 6C   04
0DA1 : 28 68 20 68 28 64 2C 74   52
0DA9 : 2C 64 30 78 30 60 34 74   E2
0DB1 : 38 64 38 68 50 60 50 68   51
0DB9 : 50 60 54 6C 54 60 58 70   67
0DC1 : 58 68 5C 68 60 68 60 74   25
0DC9 : 60 6C 5C 74 5C 78 48 78   A1
0DD1 : 62 78 48 A2 48 6C 48 6C   B8
0DD9 : 54 6C 48 6C 54 70 48 70   CE
0DE1 : 58 74 48 74 5C 6C 38 6C   F7
0DE9 : 44 6C 44 74 44 78 44 80   1D
0DF1 : 44 70 38 70 3C 70 3C 78   B3
0DF9 : 3C 70 40 88 40 78 38 7C   30
0E01 : 40 7C 38 88 38 80 3C 8C   30
0E09 : 3C 8C 38 8C 44 88 44 8C   DE
0E11 : 44 84 44 84 48 78 34 7C   4B
0E19 : 34 7C 2C 7C 34 7C 2C 80   FF
0E21 : 2C 80 30 80 34 80 34 8C   DB
0E29 : 34 54 2C 54 30 80 34 8C   0E
0E31 : 34 78 28 78 30 78 28 88   33
0E39 : 28 84 28 84 30 88 2C 88   47
0E41 : 30 70 20 70 28 74 24 74   5F
0E49 : 2C 70 20 78 20 74 24 7C   F4
0E51 : 24 74 24 74 2C 78 1C 78   2F
0E59 : 20 7C 18 7C 24 70 1C 74   6C
0E61 : 1C 70 08 70 14 70 14 74   C3
0E69 : 1C 6C 04 80 04 74 08 74   B9
0E71 : 10 78 08 78 18 78 08 80   35
0E79 : 08 84 04 84 0C 7C 0C 84   33
0E81 : 0C 7C 0C 7C 14 7C 14 80   D4
0E89 : 14 80 18 80 28 80 10 88   CB
0E91 : 10 84 10 84 24 84 24 8C   88
0E99 : 24 88 14 88 20 88 08 88   8F
0EA1 : 10 8C 0C 8C 34 8C 0C 9C   9D
0EA9 : 14 88 04 9E 04 88 08 9E   B8
0EB1 : 08 94 10 98 10 90 10 90   01
0EB9 : 28 90 14 9C 14 94 18 94   31
0EC1 : 2C 90 2C 94 2C 90 30 90   FC
0EC9 : 38 90 34 9C 34 98 24 98   B4
0ED1 : 30 94 30 98 30 9C 18 9C   EC
0ED9 : 1C 98 18 98 20 98 20 9C   DB
0EE1 : 34 94 38 94 3C 90 3C 94   62
0EE9 : 3C 90 40 98 40 98 38 98   6B
0EF1 : 40 9C 38 9C 44 90 44 9C   34
0EF9 : 44 FF 13 A0 FF 00 EF 00   D5
0F01 : FF 0A FF 00 FF 00 FF 6F   E4
0F09 : 5F 00 FF 00 FF 00 FF 00   68
0F11 : FF 00 EF 00 FF 00 FF 00   0C
0F19 : FF 00 FF 00 FF 6F FF 00   94
0F21 : FF 00 FF 00 4F 00 FF 6F   F4
0F29 : FF 00 FF 00 FF 00 FF 00   28
0F31 : 7E 00 FF 00 FF 0B 7F 6F   E4
0F39 : FF FF 00 FF 02 FF 00 FF   58
0F41 : 80 FF FE FE 82 FE 8B FF   AE
0F49 : 7A FF 82 FF 82 FF 82 FF   95
0F51 : 82 FF 82 FF 02 FF 00 CF   33
0F59 : 03 FF 82 FF 82 F6 FF FF   DC
0F61 : 82 FF 82 FF 00 FF 00 DA   39
0F69 : CF 04 FF 00 FF 02 FF 00   4A
0F71 : FF 10 CF 03 FF 00 FE 34   31
0F79 : FF 00 FF 00 CF 03 FF 00   8D
0F81 : FF 00 FF 00 FF 00 FF EF   60
0F89 : FF 00 FF 00 FF 00 FF 00   88
0F91 : CF 03 FF 00 FF 00 FF 00   E2
0F99 : FF 00 FF 00 FF 00 FF 00   98
0FA1 : FF 00 CF 03 FF 00 FF 00   F5
0FA9 : FF 00 FF 00 FF 00 FF CB   40
0FB1 : FF 00 FF 00 FF CB FF 00   0F
0FB9 : 10 FF 6F FD 00 FF 80 FF   66
0FC1 : 80 CD EF FD CB FD 80 CD   2E
0FC9 : 00 FD 80 FD 80 FD 80 FF   A2
0FD1 : 80 FD 80 FF 80 FF 80 EF   5A
0FD9 : 80 FD 80 FD 2B ED 80 FD   58
0FE1 : EF FD 80 FD 80 FF EF 6F   55
0FE9 : 00 80 80 FF 80 FF 80 FF   53
0FF1 : 80 EF 00 2F 00 FF 01 CB   EA
0FF9 : 00 FF 00 FF 00 EF 00 FF   78
1001 : 0A FF 00 FF 00 FF 6F 7F   C8
1009 : 00 FF 00 FF 00 FF 00 FF   09
1011 : 00 EF 00 FF 00 FF 00 FF   09
1019 : 00 FF 00 FF 6F FF 00 FF   10
1021 : 00 FF 00 6F 00 FF 6F FF   CD
1029 : 00 FF 00 FF 00 FF 00 7E   26
1031 : 00 FF 00 FF 0B 7F 6F FF   9B
1039 : FF 00 FE 02 FF 00 7F 00   36
1041 : 7F 7E FE 02 7E 0B 7F 7A   32
1049 : FF 02 7F 02 7F 02 7F 02   74
1051 : 7F 02 7F 02 7F 00 FF FF   E9
1059 : 7F 02 7F 02 F6 FF 7F 02   6B
1061 : 7F 02 7F 00 7F 00 FA FF   A5
1069 : FF 7F 7F 00 7F 02 FF 00   10
1071 : FF 10 CF 03 FF 00 FE 34   31
1079 : FF 00 FF 00 CF 03 FF 00   8D
1081 : FF 00 FF 00 FF 02 FE EF   6C
1089 : FF 00 FF 00 FF 00 FF 00   88
1091 : FF FE FF 00 FF 00 FF 00   0F
1099 : FF 00 FF 00 FE 00 FF 00   88
10A1 : FF 00 CF 03 FF 00 FF 00   F5
10A9 : FF 00 FF 00 FF 00 FF CB   40
10B1 : FF 00 FF 00 FF CB FE 00   0B
10B9 : 10 FF 6F FD 00 FF 80 FF   66
10C1 : 80 CD EF FD CB FD 80 CD   2E
10C9 : 00 FD 80 FD 80 FD 80 FF   A2
10D1 : 80 FD 80 FF 80 FF 80 CF   19
10D9 : 01 CF 80 FD 80 FD 2B CD   E2
10E1 : 80 FD EF FD 80 FD 80 FF   15
10E9 : CF 01 CF 6F 00 80 80 FF   21
10F1 : 80 FF 80 FF 80 EF 00 0F   36
10F9 : 00 FF 01 DB 00 FF 00 FF   B5
1101 : 00 EF 00 FF 0A FF 00 FF   99
1109 : 00 FF 6F 5F 00 FF 00 FF   D1
1111 : 00 FF 00 FF 00 CF 01 CF   33
1119 : 00 FF 00 FF 00 FF 00 FF   19
1121 : 00 FF 6F FF 00 FF 00 FF   FD
1129 : 00 4F 00 FF 6F FF 00 FF   C8
1131 : 00 FF 00 FF 00 7E 00 FF   25
1139 : 00 FF 0B 7E 6F FF FF 00   E3
1141 : FF 02 FF 00 7F 00 7F 7E   34
1149 : FE 02 7E 0A 7F 7A FF 02   F9
1151 : 7F 02 7F 02 7F 02 7F 02   FC
1159 : 7F 02 7F 00 FF FF 7F 02   BB
1161 : 7F 02 F6 FF 7F 02 7F 02   A9
1169 : 7F 00 7F 00 DA FF FF 7F   75
1171 : 7F 00 7F 02 FF 00 FF 10   31
1179 : CF 03 FF 00 FE 34 FF 00   5B
1181 : FF 00 CF 03 FF 00 FF 00   D5
1189 : FF 00 FF 00 FF EF FF 00   08
1191 : FF 00 FF 00 FF 00 CF 03   D5
1199 : FF 00 FF 00 FF 00 FF 00   98
11A1 : FF 00 FF 00 FF 00 FF 00   A0
11A9 : CF 03 FF 00 FF 00 FF 00   FA
11B1 : FF 00 FF 00 FF CB FF 00   0F
11B9 : FF 00 FF CB FF 00 10 FF   72
11C1 : 6F FD 00 FF 00 7F 00 4D   C5
11C9 : 6F 7D 4B 7D 00 4D 00 7D   DF
11D1 : 00 7D 00 7D 00 7F 00 7D   37
11D9 : 00 7F 00 7F 00 EF 00 7D   03
11E1 : 00 7D 2B 6D 00 7D 6F 7D   BD
11E9 : 00 7D 00 FF EF 6F BF 03   27
11F1 : 7F 00 7F 00 FF 00 EF 00   10
11F9 : 2F 00 FF 01 CB 00 FF 00   05
1201 : FF 00 6F 00 FF 0A FF 00   2C
1209 : FF 00 FF 6F 7F 00 FF 00   EE
1211 : FF 00 FF 00 FF 00 4F 00   4D
1219 : FF 00 FF 00 FF 00 FF 00   18
1221 : FF 6F FF 00 FF 00 FF 00   D8
1229 : 6F 00 FF 6F FF 00 FF 00   86
1231 : FF 00 FF 00 7E 00 FF 00   18
1239 : FF 0B 7F 6F FF FF 00 FE   8A
1241 : 02 FF 00 7F 00 7F 7E FE   27
1249 : 02 7E 0A 7F 7A FF 02 7F   AB
1251 : 02 7F 02 7F 02 7F 02 7F   A6
1259 : 02 7F 00 FF FF 7F 02 7F   1E
1261 : 02 F6 FF 7F 02 7F 02 7F   F1
1269 : 00 7F 00 FA FF FF 7F 7F   B5
1271 : 00 7F 02 FF 00 FF 10 CF   91
1279 : 03 FF 00 FE 34 FF 00 FF   9F
1281 : 00 CF 03 FF 00 FF 00 FF   2A
1289 : 00 FF 02 FE EF FF 00 FF   E8
1291 : 00 FF 00 FF 00 FF FE FF   8D
1299 : 00 FF 00 FF 00 CF 00 AA   6D
12A1 : CF 00 AA CF 00 AA CF 00   A9
12A9 : AA CF 00 AA CF 00 AA CF   D7
12B1 : 00 AA CF 00 AA CF 00 AA   79
12B9 : CF 00 AA CF 00 AA CF 00   C1
12C1 : AA CF 00 AA CF 00 AA CF   EF
12C9 : 00 AA CF 00 AA CF 00 AA   91
12D1 : CF 00 AA CF 00 AA CF 00   D9
12D9 : AA CF 00 AA CF 00 AA CF   07
12E1 : 00 AA CF 00 AA CF 00 AA   A9
12E9 : CF 00 AA CF 00 AA CF 00   F1
12F1 : AA CF 00 AA CF 00 AA CF   1F
12F9 : 00 AA CF 00 AA CF 00 AA   C1
1301 : CF 00 AA CF 00 AA CF 00   09
1309 : AA CF 00 AA CF 00 AA CF   37
1311 : 00 AA CF 00 AA CF 00 AA   D9
1319 : CF 00 AA CF 00 AA CF 00   21
1321 : AA CF 00 AA CF 00 AA CF   4F
1329 : 00 AA CF 00 AA CF 00 AA   F1
1331 : CF 00 AA CF 00 AA CF 00   39
1339 : AA CF 00 AA CF 00 AA CF   67
1341 : 00 AA CF 00 AA CF 00 AA   09
1349 : CF 00 AA CF 00 AA CF 00   51
1351 : AA CF 00 AA CF 00 AA CF   7F
1359 : 00 AA CF 00 AA CF 00 AA   21
1361 : CF 00 AA CF 00 AA CF 00   69
1369 : AA CF 00 AA CF 00 AA CF   97
1371 : 00 AA CF 00 AA CF 00 AA   39
1379 : CF 00 AA CF 00 AA CF 00   81
1381 : AA CF 00 AA CF 00 AA CF   AF
1389 : 00 AA CF 00 AA CF 00 AA   51
1391 : CF 00 AA CF A0 AA 48 8A   9A
1399 : 0A 0A 8D 23 C9 68 4A 8D   94
13A1 : 24 C9 B0 37 A2 00 A9 15   B8
13A9 : 8D 25 C9 AD 24 C9 8D 20   F8
13B1 : C9 20 B5 C0 B1 3F 9D 00   9B
13B9 : 64 EE 20 C9 E8 20 B5 C0   BD
13C1 : B1 3F 9D 00 64 EE 20 C9   4B
13C9 : E8 20 B5 C0 B1 3F 9D 00   D2
13D1 : 64 E8 EE 23 C9 CE 25 C9   05
13D9 : D0 D1 60 A9 15 8D 25 C9   C5
13E1 : A2 00 AD 24 C9 8D 20 C9   90
13E9 : 20 B5 C0 B1 3F CF 04 0A   E1
13F1 : 9D 00 64 EE 20 C9 20 B5   C1
13F9 : C0 B1 3F 48 CF 04 4A 18   E1
1401 : 7D 00 64 9D 00 64 E8 68   E3
1409 : CF 04 0A 9D 00 64 EE 20   30
1411 : C9 20 B5 C0 B1 3F 48 CF   46
1419 : 04 4A 18 7D 00 64 9D 00   92
1421 : 64 E8 68 CF 04 0A 9D 00   14
1429 : 64 EE 20 C9 20 B5 C0 B1   5C
1431 : 3F CF 04 4A 18 7D 00 64   D8
1439 : 9D 00 64 E8 EE 23 C9 CE   D9
1441 : 25 C9 D0 98 60 A9 00 8D   00
1449 : 22 C9 AD 20 C9 CF 03 0A   FA
1451 : 8D 21 C9 2E 22 C9 20 73   7F
1459 : C2 A0 00 60 AD 00 DD 29   1C
1461 : FE 8D 00 DD AD 48 03 8D   26
1469 : 20 D0 A9 3B 8D 11 D0 A9   BB
1471 : 80 8D 18 D0 A9 40 85 40   0B
1479 : A9 00 85 3F A0 00 91 3F   3A
1481 : C8 D0 FB E6 40 A6 40 E0   89
1489 : 60 D0 F3 A0 00 AD 48 03   F7
1491 : 91 3F C8 D0 FB E6 40 A6   53
1499 : 40 E0 64 D0 F3 A9 03 8D   30
14A1 : 15 D0 8D 17 D0 8D 1D D0   F4
14A9 : A2 40 A9 FF 9D 40 64 CA   D9
14B1 : D0 FA 20 51 C1 AD 4A 03   E9
14B9 : 48 29 0F 8D 28 D0 68 CF   56
14C1 : 04 4A 8D 27 D0 A9 19 8D   0C
14C9 : 00 D0 8D 02 D0 A9 32 8D   13
14D1 : 01 D0 8D 03 D0 A9 00 8D   74
14D9 : 10 D0 A9 00 AA 4C 00 C0   4A
14E1 : A9 65 85 FE A9 00 85 FD   2B
14E9 : 20 77 C1 8D 26 C9 20 77   07
14F1 : C1 8D 27 C9 20 77 C1 8D   5C
14F9 : 28 C9 20 77 C1 8D 29 C9   BE
1501 : 20 FA C1 4C 59 C1 A0 00   BE
1509 : B1 FD C9 FF F0 08 E6 FD   12
1511 : F0 01 60 E6 FE 60 68 68   DC
1519 : A9 02 8D 2A C9 8D 2B C9   B5
1521 : AD 2A C9 8D 26 C9 8D 28   3F
1529 : C9 AD 2B C9 8D 27 C9 8D   21
1531 : 29 C9 20 FA C1 AD 2B C9   70
1539 : 18 69 04 8D 2B C9 C9 66   AE
1541 : D0 DE A9 02 8D 2B C9 AD   E0
1549 : 2A C9 18 69 04 8D 2A C9   74
1551 : C9 7A D0 CC AD 2A C9 8D   93
1559 : 26 C9 8D 28 C9 AD 2B C9   16
1561 : 8D 27 C9 8D 29 C9 20 FA   FD
1569 : C1 AD 2B C9 18 69 04 8D   FD
1571 : 2B C9 C9 4A D0 DE A9 02   EB
1579 : 8D 2B C9 AD 2A C9 18 69   E8
1581 : 04 8D 2A C9 C9 A2 D0 CC   9E
1589 : 60 AD 26 C9 8D 21 C9 A9   DF
```

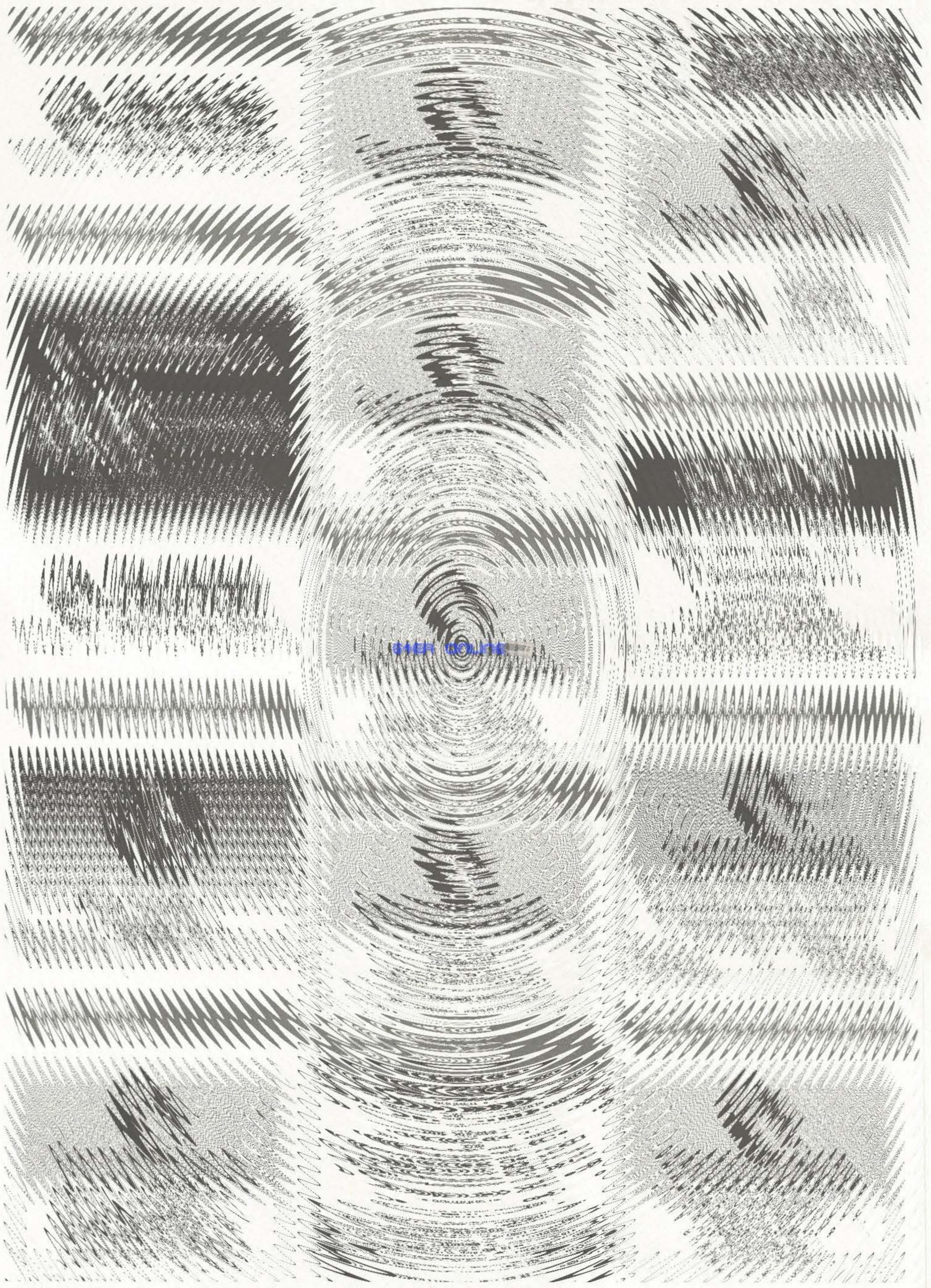
MSE-Listing »Pac-Man«. Bitte verwenden Sie bei der Eingabe den MSE (Eingabehinweise auf Seite 54).

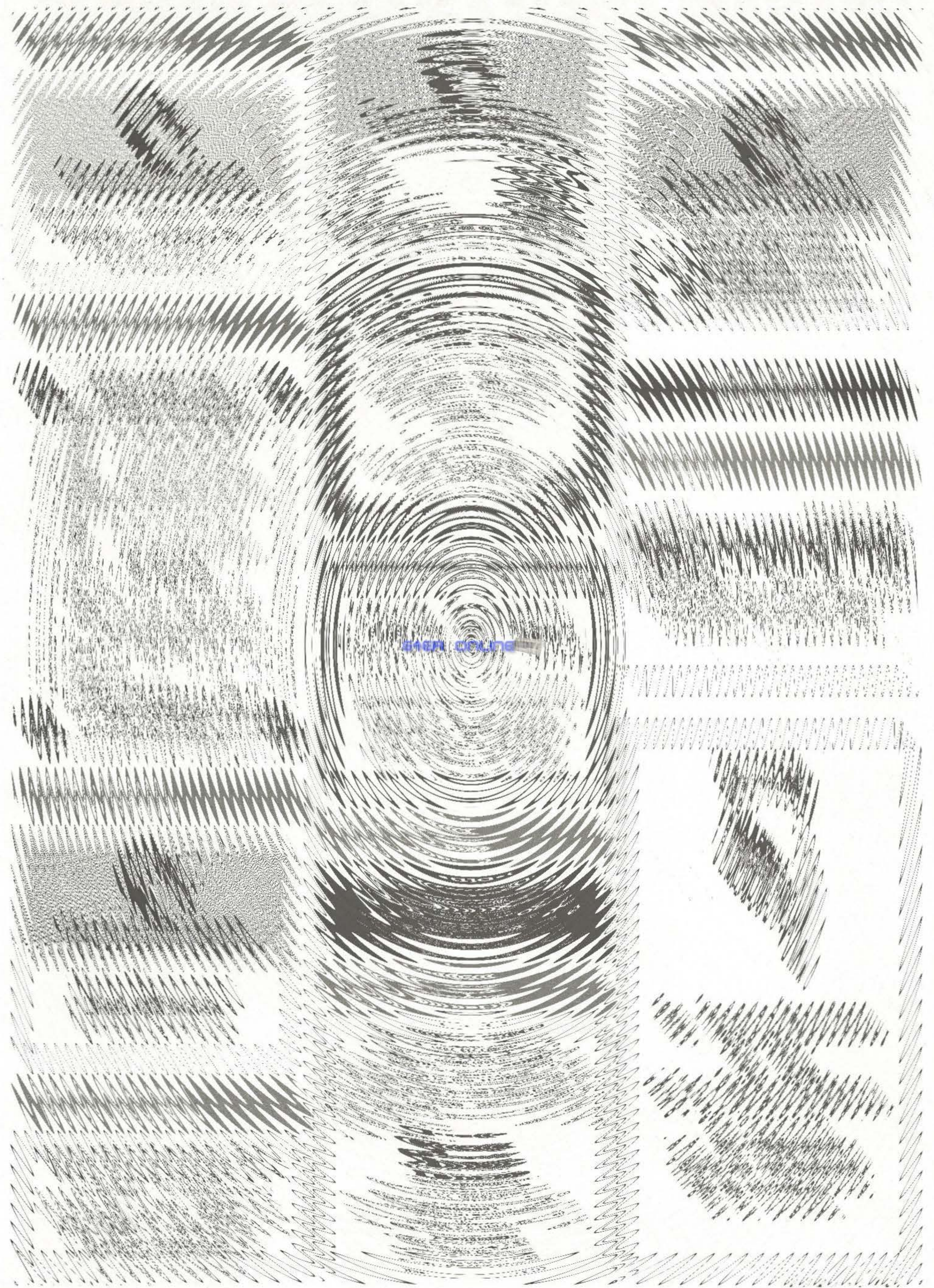


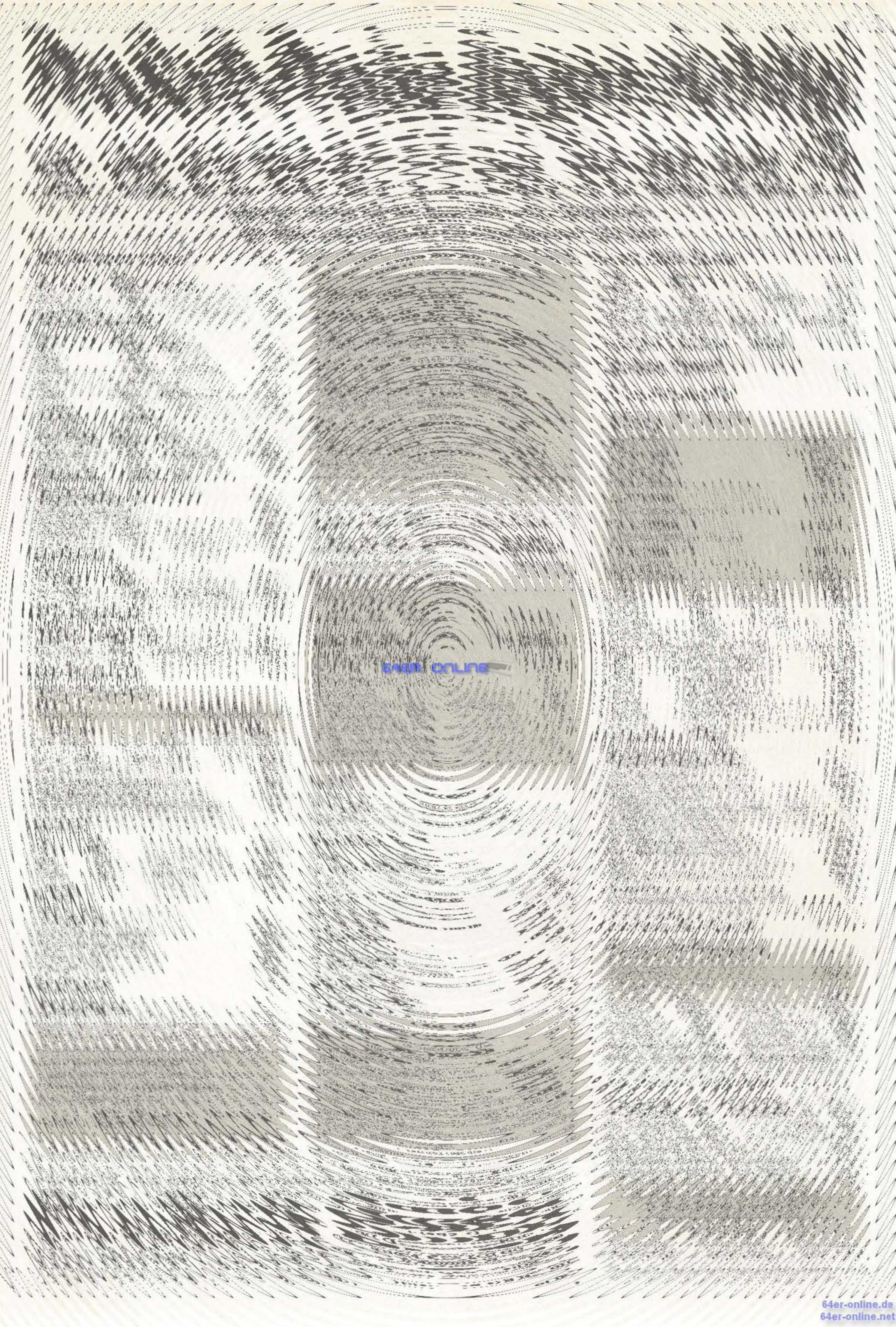
```
1591 : 00 8D 22 C9 AD 27 C9 8D  70
1599 : 23 C9 20 56 C2 A9 C4 38  71
15A1 : ED 27 C9 8D 23 C9 20 56  F4
15A9 : C2 A9 01 8D 22 C9 A9 3C  C1
15B1 : 38 ED 26 C9 B0 03 CE 22  45
15B9 : C9 8D 21 C9 20 56 C2 AD  E5
15C1 : 27 C9 8D 23 C9 20 56 C2  11
15C9 : AD 26 C9 CD 28 C9 F0 06  56
15D1 : EE 26 C9 4C FA C1 AD 27  91
15D9 : C9 CD 29 C9 D0 01 60 EE  81
15E1 : 27 C9 4C FA C1 20 63 C2  8F
15E9 : A0 00 B1 3F 0D 2C C9 91  5A
15F1 : 3F 60 20 73 C2 AD 21 C9  88
15F9 : 29 07 AA BD 32 C9 8D 2C  08
1601 : C9 60 AD 23 C9 29 F8 4A  28
1609 : 4A A8 B9 3C C9 85 3F B9  D6
1611 : 3D C9 85 40 AD 23 C9 29  0A
1619 : 07 18 65 3F 85 3F A9 40  E7
1621 : 65 40 85 40 AD 21 C9 29  6D
1629 : F8 18 65 3F 85 3F AD 22  BC
1631 : C9 65 40 85 40 60 48 8A  AB
1639 : 0A 0A 8D 23 C9 EE 23 C9  44
1641 : AE 31 C9 68 4A 8D 20 C9  2C
1649 : 90 27 20 0C C3 B1 3F 1D  F7
1651 : C0 C9 E8 91 3F EE 23 C9  EE
1659 : 20 0C C3 B1 3F 1D C0 C9  1A
1661 : E8 91 3F EE 23 C9 20 0C  D9
1669 : C3 B1 3F 1D C0 C9 91 3F  97
1671 : 60 20 0C C3 CF 03 E8 B1  79
1679 : 3F 1D C0 C9 E8 91 3F EE  A6
1681 : 23 C9 20 0C C3 B1 3F 1D  13
1689 : C0 C9 E8 91 3F EE 23 C9  26
1691 : 20 0C C3 B1 3F 1D C0 C9  52
1699 : 91 3F 60 A9 00 8D 22 C9  A0
16A1 : AD 20 C9 CF 03 0A 8D 21  C4
16A9 : C9 2E 22 C9 20 73 C2 A0  35
16B1 : 00 60 48 8A 0A 0A 8D 23  B2
16B9 : C9 68 4A 8D 20 C9 B0 12  32
16C1 : A2 03 EE 23 C9 20 0C C3  5A
16C9 : B1 3F 29 8F 91 3F CA D0  36
16D1 : F1 60 A2 03 EE 23 C9 20  6B
16D9 : 0C C3 B1 3F 29 F8 91 3F  3A
16E1 : CA D0 F1 60 48 8A 0A 0A  B1
16E9 : 8D 23 C9 EE 23 C9 68 4A  0F
16F1 : 8D 20 C9 90 22 20 0C C3  EE
16F9 : B1 3F 09 02 91 3F EE 23  E2
1701 : C9 20 0C C3 B1 3F 09 05  99
1709 : 91 3F EE 23 C9 20 0C C3  AF
1711 : B1 3F 09 07 91 3F 60 4C  B2
1719 : 9D C4 48 8A 0A 0A 8D 23  E9
1721 : C9 68 4A 8D 20 C9 B0 25  C0
1729 : EE 23 C9 20 0C C3 B1 3F  43
1731 : 29 8F 91 3F EE 23 C9 20  DD
1739 : 0C C3 B1 3F 29 AF 91 3F  50
1741 : EE 23 C9 20 0C C3 B1 3F  5B
1749 : 29 8F 91 3F 60 EE 23 C9  24
1751 : 20 0C C3 B1 3F 29 F8 91  E3
1759 : 3F EE 23 C9 20 0C C3 B1  E6
1761 : 3F 29 FA 91 3F EE 23 C9  B1
1769 : 20 0C C3 B1 3F 29 F8 91  FB
1771 : 3F 60 20 63 C2 A0 00 B1  E9
1779 : 3F 2D 2C C9 8D 3F 03 60  33
1781 : A9 00 8D 2D C9 8D 2E C9  89
1789 : 8D 2F C9 8D 30 C9 A9 A8  1B
1791 : 8D 21 C9 AD 49 03 A2 05  18
1799 : 9D 40 62 CA D0 FA A9 00  D4
17A1 : 20 70 C4 4C 47 C4 EE 2D  65
17A9 : C9 AD 2D C9 C9 0A D0 47  8C
17B1 : A9 00 8D 2D C9 EE 2E C9  C4
17B9 : AD 2E C9 C9 0A D0 2D A9  58
17C1 : 00 8D 2E C9 EE 2F C9 AD  37
17C9 : 2F C9 C9 0A D0 13 A9 00  DD
17D1 : 8D 2F C9 EE 30 C9 A9 88  4F
17D9 : 8D 21 C9 AD 30 C9 20 70  D2
17E1 : C4 A9 90 8D 21 C9 AD 2F  C5
17E9 : C9 20 70 C4 A9 98 8D 21  4F
17F1 : C9 AD 2E C9 20 70 C4 A9  42
17F9 : A0 8D 21 C9 AD 2D C9 A2  92
1801 : 00 8E 22 C9 A2 70 8E 23  38
1809 : C9 CF 03 0A AA 20 73 C2  BB
1811 : BD 70 C9 A0 00 91 3F EE  F4
1819 : 23 C9 E8 AD 23 C9 C9 78  A9
1821 : D0 EB A9 32 8D 01 D4 A2  11
1829 : 0A 4C F8 C8 20 0C C3 B1  85
1831 : 3F 09 20 91 3F EE 23 C9  BB
1839 : 20 0C C3 B1 3F 09 50 91  27
1841 : 3F EE 23 C9 20 0C C3 B1  CE
1849 : 3F 09 70 91 3F 60 AD 00  09
1851 : DC 48 AD 19 C9 AE 1A C9  EE
1859 : 20 23 C3 68 C9 7B D0 03  CA
1861 : 4C 84 C5 C9 7E D0 03 4C  AD
1869 : F4 C5 C9 77 D0 03 4C BB  6F
1871 : C5 C9 7D D0 03 4C 3A C6  9D
1879 : AD 19 C9 AE 1A C9 20 A7  BB
1881 : C2 AD 19 C9 0A 0A 8D 1B  F7
1889 : C9 A9 00 69 00 8D 1C C9  C5
1891 : AD 1B C9 18 69 06 8D 1B  74
1899 : C9 AD 1C C9 69 00 8D 1C  7E
18A1 : C9 AD 1A C9 0A 0A 18 69  25
18A9 : 20 8D 1D C9 AE 19 C9 CA  81
18B1 : CA 8E 1E C9 AE 1A C9 CA  FC
18B9 : CA 8E 1F C9 AD 19 C9 C9  2A
18C1 : 03 B0 03 20 82 C6 AD 1A  2A
18C9 : C9 C9 03 B0 03 20 92 C6  57
18D1 : AD 19 C9 C9 4A 90 03 20  2C
18D9 : 9D C6 AD 1A C9 C9 2C 90  45
18E1 : 03 20 AD C6 AD 1E C9 AE  89
18E9 : 1F C9 20 00 C0 AD 1D C9  76
18F1 : 8D 01 D0 8D 03 D0 AD 1C  8A
18F9 : C9 0A 18 6D 1C C9 8D 10  E1
1901 : D0 AD 1B C9 8D 00 D0 8D  DF
1909 : 02 D0 20 9C C7 20 0E C9  58
1911 : 4C CF 01 BF C4 AD 1A C9  33
1919 : 0A 0A AA E8 8E 23 C9 A9  6C
1921 : 00 8D 22 C9 AD 19 C9 0A  88
1929 : 0A 8D 21 C9 2E 22 C9 20  D7
1931 : E3 C3 A9 0C 8D 31 C9 AD  C7
1939 : 3F 03 F0 03 4C E9 C4 AE  1B
1941 : 19 C9 CA 8E 19 C9 20 B8  95
1949 : C6 4C E9 C4 AD 1A C9 0A  2F
1951 : 0A AA E8 8E 23 C9 A9 00  E3
1959 : 8D 22 C9 AE 19 C9 E8 8A  D8
1961 : 0A 0A 8D 21 C9 2E 22 C9  22
1969 : 20 E3 C3 A9 00 8D 31 C9  66
1971 : AD 3F 03 F0 03 4C E9 C4  60
1979 : AE 19 C9 E8 8E 19 C9 20  5C
1981 : F2 C6 4C E9 C4 AD 1A C9  DC
1989 : 0A 0A 8D 23 C9 A9 00 8D  65
1991 : 22 C9 AD 19 C9 0A 0A 8D  57
1999 : 21 C9 2E 22 C9 AD 21 C9  91
19A1 : 18 69 01 8D 21 C9 AD 22  BB
19A9 : C9 69 00 8D 22 C9 20 E3  91
19B1 : C3 A9 12 8D 31 C9 AD 3F  16
19B9 : 03 F0 03 4C E9 C4 AE 1A  32
19C1 : C9 CA 8E 1A C9 20 2A C7  AC
19C9 : 4C E9 C4 AE 1A C9 E8 8A  B9
19D1 : 0A 0A 8D 23 C9 A9 00 8D  AD
19D9 : 22 C9 AD 19 C9 0A 0A 8D  9F
19E1 : 21 C9 2E 22 C9 AD 21 C9  D9
19E9 : 18 69 01 8D 21 C9 AD 22  03
19F1 : C9 69 00 8D 22 C9 20 E3  D9
19F9 : C3 A9 06 8D 31 C9 AD 3F  5B
1A01 : 03 F0 03 4C E9 C4 AE 1A  7A
1A09 : C9 E8 8E 1A C9 20 64 C7  EC
1A11 : 4C E9 C4 A9 00 8D 1C C9  29
1A19 : A9 19 8D 1B C9 A9 00 8D  1B
1A21 : 1E C9 60 A9 32 8D 1D C9  09
1A29 : A9 00 8D 1F C9 60 A9 01  62
1A31 : 8D 1C C9 A9 2E 8D 1B C9  C3
1A39 : A9 4A 8D 1E C9 60 A9 D0  16
1A41 : 8D 1D C9 A9 2C 8D 1F C9  44
1A49 : 60 A9 00 8D 22 C9 AE 19  8D
1A51 : C9 E8 8A 0A 0A 8D 21 C9  97
1A59 : 2E 22 C9 AD 21 C9 38 E9  D5
1A61 : 02 8D 21 C9 AD 22 C9 E9  92
1A69 : 00 8D 22 C9 AD 1A C9 0A  D8
1A71 : 0A AA E8 E8 8E 23 C9 20  91
1A79 : E3 C3 AD 3F 03 F0 03 20  95
1A81 : 17 C4 60 A9 00 8D 22 C9  D0
1A89 : AD 19 C9 0A 0A 8D 21 C9  9C
1A91 : 2E 22 C9 AD 21 C9 18 69  8C
1A99 : 02 8D 21 C9 AD 22 C9 69  C9
1AA1 : 00 8D 22 C9 AD 1A C9 0A  10
1AA9 : 0A AA E8 E8 8E 23 C9 20  C9
1AB1 : E3 C3 AD 3F 03 F0 03 20  CD
1AB9 : 17 C4 60 A9 00 8D 22 C9  08
1AC1 : AD 17 C9 0A 0A 8D 21 C9  D4
1AC9 : 2E 22 C9 AD 21 C9 18 69  C4
1AD1 : 02 8D 21 C9 AD 22 C9 69  01
1AD9 : 00 8D 22 C9 AE 1A C9 E8  16
1AE1 : 8A 0A 0A AA CA CA 8E 23  CC
1AE9 : C9 20 E3 C3 AD 3F 03 F0  F6
1AF1 : 03 20 17 C4 60 A9 00 8D  D1
1AF9 : 22 C9 AD 19 C9 0A 0A 8D  BF
1B01 : 21 C9 2E 22 C9 AD 21 C9  F9
1B09 : 18 69 02 8D 21 C9 AD 22  63
1B11 : C9 69 00 8D 22 C9 AD 1A  9C
1B19 : C9 0A 0A AA E8 E8 8E 23  15
1B21 : C9 20 E3 C3 AD 3F 03 F0  2E
1B29 : 03 20 17 C4 60 A0 00 8C  BF
1B31 : 26 C9 B9 F0 C9 AA B9 E8  73
1B39 : C9 8E 3B C9 8D 3A C9 20  63
1B41 : 8B C3 20 D3 C7 AC 26 C9  3E
1B49 : AD 3B C9 99 F0 C9 AA AD  9D
1B51 : 3A C9 99 E8 C9 20 55 C3  6E
1B59 : EE 26 C9 AC 26 C9 CC 4B  DD
1B61 : 03 D0 CF 01 CF 60 AD 19  C9
1B69 : C9 CD 3A C9 F0 18 90 03  F9
1B71 : 4C 26 C8 4C 55 C8 AD 1A  12
1B79 : C9 CD 3B C9 F0 3A 90 03  5A
1B81 : 4C 82 C8 4C BE C8 AD 1A  E7
1B89 : C9 CD 3B C9 D0 E8 A2 00  20
1B91 : 8E 22 C9 A2 60 8E 23 C9  92
1B99 : A2 88 8E 21 C9 20 73 C2  38
1BA1 : A0 00 B9 FC C9 F0 05 91  AB
1BA9 : 3F C8 D0 F6 AD 00 DC C9  41
1BB1 : 6F D0 F9 CF 04 68 60 60  C6
1BB9 : AD 3B C9 0A 0A AA E8 8E  6E
1BC1 : 23 C9 A9 00 8D 22 C9 AE  A2
1BC9 : 3A C9 E8 8A 0A 0A 8D 21  DD
1BD1 : C9 2E 22 C9 20 E3 C3 AD  FF
1BD9 : 3F 03 F0 03 4C E3 C7 AE  97
1BE1 : 3A C9 E8 8E 3A C9 60 AD  DB
1BE9 : 3B C9 0A 0A AA E8 8E 23  3F
1BF1 : C9 A9 00 8D 22 C9 AD 3A  DC
1BF9 : C9 0A 0A 8D 21 C9 2E 22  59
1C01 : C9 20 E3 C3 AD 3F 03 F0  0E
1C09 : 03 4C E3 C7 AE 3A C9 CA  9E
1C11 : 8E 3A C9 60 AE 3B C9 E8  F8
1C19 : 8A 0A 0A 8D 23 C9 A9 00  04
1C21 : 8D 22 C9 AD 3A C9 0A 0A  16
1C29 : 8D 21 C9 2E 22 C9 AD 21  E8
1C31 : C9 18 69 01 8D 21 C9 AD  E5
1C39 : 22 C9 69 00 8D 22 C9 20  EC
1C41 : E3 C3 AD 3F 03 F0 01 60  D6
1C49 : AE 3B C9 E8 8E 3B C9 60  CF
1C51 : AD 3B C9 0A 0A 8D 23 C9  7D
1C59 : A9 00 8D 22 C9 AD 3A C9  30
1C61 : 0A 0A 8D 21 C9 2E 22 C9  22
1C69 : AD 21 C9 18 69 01 8D 21  33
1C71 : C9 AD 22 C9 69 00 8D 22  E4
1C79 : C9 20 E3 C3 AD 3F 03 F0  86
1C81 : 01 60 AE 3B C9 CA 8E 3B  69
1C89 : C9 60 A0 00 A9 81 8D 04  8F
1C91 : D4 88 D0 F8 CA D0 F3 A9  53
1C99 : 80 8D 04 D4 88 D0 F8 60  2F
1CA1 : A2 32 A0 00 88 D0 FD CA  21
1CA9 : D0 F8 60 BF 19 80 40 20  DC
1CB1 : 10 08 04 02 01 BF 04 40  A5
1CB9 : 01 80 02 C0 03 00 05 40  58
1CC1 : 06 80 07 C0 08 00 0A 40  0A
1CC9 : 0B 80 0C C0 0D 00 0F 40  BD
1CD1 : 10 80 11 C0 12 00 14 40  70
1CD9 : 15 80 16 C0 17 00 19 40  22
1CE1 : 1A 80 1B C0 1C 00 1E 40  D5
1CE9 : 1F 7E CF 06 66 7E CF 08  A6
1CF1 : 06 7E 06 06 7E CF 03 60  AC
1CF9 : 7E 7E 06 06 7E CF 03 06  77
1D01 : 7E CF 03 66 7E CF 04 06  77
1D09 : 7E 60 60 7E CF 03 06 7E  C9
1D11 : 7E 60 60 7E CF 03 66 7E  53
1D19 : 7E CF 07 06 7E 66 66 7E  B3
1D21 : CF 03 66 7E 7E 66 66 7E  8D
1D29 : CF 03 06 7E 03 05 03 30  90
1D31 : 40 30 02 05 05 20 50 50  DE
1D39 : 06 01 06 60 50 60 05 05  74
1D41 : 02 50 50 20 14 28 14 28  A7
1D49 : 0A 32 0A 32 0F 0F 23 23  71
1D51 : 0A 0A 28 28 14 28 14 28  93
1D59 : 0A 32 0A 32 0F 0F 23 23  81
1D61 : 0A 0A 28 28 BF 04 71 8A  76
1D69 : 82 B3 CF 03 8A 72 C5 2A  C1
1D71 : 2A E8 CF 04 28 3C A0 A0  AC
1D79 : B8 CF 03 A0 BC 72 CF 05  96
1D81 : 8A 89 70 2F 28 28 2E 28  9F
1D89 : 28 48 8F 78 44 44 78 60  D1
1D91 : 50 48 44 BF F4 A9 00 8D  C6
1D99 : 19 C9 8D 1A C9 4C CF 01  7E
1DA1 : BF C4 BF 13 20 CC C0 A0  C1
1DA9 : 08 B9 E0 C9 99 F0 C9 B9  BB
1DB1 : D8 C9 99 E8 C9 88 D0 F1  F9
1DB9 : A9 00 8D 05 D4 A9 F0 8D  E0
1DC1 : 06 D4 A9 0F 8D 18 D4 4C  03
1DC9 : F1 C3 CF 00 AA CF 00 AA  0E
1DD1 : CF 00 AA CF 00 AA CF 9B  11
1DD9 : AA AA AA AA AA AA AA AA  D8
```

MSE-Listing »Pac-Man« (Schluß). Bitte beachten Sie die Eingabehinweise auf Seite 54.









Cross-Ref optimiert

Viele Verbesserungsvorschläge sind uns seit Erscheinen der Ausgabe 6/85 zu dem Dokumentationsprogramm »Cross-Ref« zugeschickt worden. Die besten sind hier zusammengestellt. Jetzt ist es zum Beispiel auch möglich, Simons Basic-Programme bearbeiten zu lassen.

Der Ärger mit dem GOTO

Zunächst muß im Programm (siehe Listing 1) darauf geachtet werden, daß Zeichen innerhalb von Anführungsstrichen grundsätzlich überlesen werden. Das gilt insbesondere in DATA- beziehungsweise REM-Zeilen. Das Flag »Y2« wird jeweils beim ersten Anführungszeichen gesetzt und beim nächsten Auftreten, spätestens am Zeilenende (X=0), gelöscht (Zeilen 342 und 345 sowie 762, 764 und 820, 825). Zeilensprünge nach »THEN« (Token 167) mit nachfolgendem »GOSUB« etc. werden in der alten Programmversion nicht gefunden. Die zusätzliche Zeile 495 testet bei vorausgegangenem »NEXT« (Flag »Y1« wurde dann in Zeile 358 gesetzt) die nachfolgenden Sprungbefehle »GOTO«, »RUN«, »GOSUB« und »GOTO« (Token 203 und 164), wobei das Flag für »GO« in Zeile 355 gesetzt wurde; der Test auf nachfolgendes »TO« erfolgt in Zeile 360.

Da beim Erstellen eines neuen Strings in der Version aus 64'er, Ausgabe 6/85, kein Test auf ein bereits vorhandenes gleiches Sprungziel erfolgt (dies geschieht nur innerhalb eines Strings in Zeile 460), bringen die neuen Zeilen 450, 455 und 458 Abhilfe; Entsprechendes gilt bei den Variablen für die Zeilen 920, 925 und 928.

Ein ganz wichtiger Punkt ist die Tatsache, daß »E« (ASCII-Code=69) nach Ziffern oder nach einem Punkt keine Variable ist. Dieses wird in Zeile 855 in Verbindung mit Zeile 90 (X2=ASCII-Code des vorausgehenden Zeichens) untersucht. Längere Variablennamen, die im Commodore-Basic vorkommen können, werden in der Programmversion falsch interpretiert. Durch die neue Zeile 952 werden maximal zwei Zeichen und/oder »$,%H,(« (Zeilen 870 beziehungsweise 880) zugelassen. Da das nachfolgende Zeichen bereits in Zeile 870 gelesen werden mußte, sorgt das Flag »Y3« dafür, daß in Zeile 750 dieses Zeichen nicht übersprungen wird (letzteres gilt auch für die Zeile 965 beim Auftreten eines Anführungszeichens).

Damit die Zeilensprünge in der richtigen Reihenfolge ausgegeben werden, sollten die Sortier-Routine (Zeilen 520 bis 540) und die Ausgabezeilen 650 und 660 entsprechend der Variablen-Ausgabe (siehe Zeilen 990 und 1 140) geändert werden.

Einige der angesprochenen Punkte werden durch das beiliegende Testprogramm (Listing 2) deutlich. Nach Eingabe der zusätzlichen beziehungsweise geänderten Zeilen dürfte einer guten Programmdokumentation nichts mehr im Wege stehen.

(Gerhard Löber/tr)

```
90 X2=X:GET#1,A$:X=ASC(A$+CHR$(0)):IF(64 A
   ND ST)=0 THEN RETURN                    <208>
325 Y=0:Y2=0                               <012>
330 GOSUB 90                               <078>
342 IF X=34 THEN Y2=NOT Y2                 <096>
345 IF Y2 THEN 330                         <255>
348 IF X=137 OR X=138 OR X=141 OR X=167 OR
     X=203 THEN S1$="":Y1=0:Y2=0:GOTO 355  <171>
355 IF X=203 THEN Y=1                      <023>
358 IF X=167 THEN Y1=1                     <150>
360 GOSUB 90:IF X=32 OR X=164 AND Y THEN 3
    60                                     <162>
450 IF LEN(PS$(I))<70 THEN 460             <137>
455 IF RIGHT$(PS$(I),5)=ZE$THEN 470        <012>
458 NEXT:PS=PS+1:I=PS:GOSUB 130:PS$(I)=S1$
    +":"+ZE$:GOTO 470                      <116>
490 IF X=58 OR X>64 AND X<91 THEN 330      <188>
495 IF Y1 THEN IF X=137 OR X=138 OR X=141
    OR X=203 THEN 355                      <083>
498 IF X>127 THEN 330                      <166>
520 FOR I=1 TO PS-1                        <108>
530 FOR J=I+1 TO PS                        <149>
540 IF LEFT$(PS$(I),5)>LEFT$(PS$(J),5)GOTO
     560                                   <045>
650 PS$(0)="":FOR I=PS TO 1 STEP-1         <065>
660 IF LEFT$(PS$(I+1),6)<>LEFT$(PS$(I),6)T
    HEN PRINT#1,PS$(I):GOTO 680            <139>
745 Y2=0:Y3=0                              <166>
750 Y1=0:IF Y3=0 THEN GOSUB 90             <255>
760 Y3=0:IF X=0 THEN 740                   <061>
762 IF X=34 THEN Y2=NOT Y2                 <006>
764 IF Y2 THEN 750                         <166>
770 IF X=131 OR X=143 THEN 800             <158>
820 IF X=34 THEN Y2=NOT Y2                 <066>
825 IF Y2 THEN 810                         <036>
855 IF X=69 THEN IF X2>47 AND X2<58 OR X2=
    46 THEN 750                            <168>
865 IF X=34 THEN Y3=1:GOTO 890             <136>
870 IF X=36 OR X=37 THEN V1$=V1$+A$:GOSUB
    90:Y3=1:GOTO 880                       <177>
875 IF X>47 AND X<58 OR X>64 AND X<91 THEN
    [illegible]52                          <255>
920 IF LEN(VA$(I))<50 THEN 930             <094>
925 IF RIGHT$(VA$(I),5)=ZE$THEN 940        <026>
928 NEXT:VA=VA+1:I=VA:GOSUB 130:VA$(I)=V1$
    +" :"+ZE$:GOTO 940                     <169>
952 IF Y1=0 THEN V1$=V1$+A$:Y1=1           <180>
954 GOTO 860                               <016>
```

© 64'er

Listing 1. Diese Zeilen müssen Sie neu eintippen. Verwenden Sie bitte dazu den neuen Checksummer (Seite 54)

```
10 REM TESTPROGRAMM FUER "XREF"            <085>
20 READ TEXT$,TEXTE$,TEX$(1),ANZAHL        <214>
30 IF AN>1E3 THEN GOSUB 1000               <180>
40 PRINT TE$AN                             <127>
50 GOTO 1000                               <246>
60 PRINT "*Ř1000Ğ*":REM <CBM-M> 1000 <CBM-
   G>                                      <133>
70 DATA "ABC,DEF","UVW:XYZ",TTT,3          <126>
1000 PRINT TE$"TEST"                       <205>
```

© 64'er

Listing 2. Dieses Beispielprogramm sollten Sie einmal vor und einmal nach den Änderungen von »Cross-Ref 64« bearbeiten lassen

Simons Basic dokumentieren

Zuerst einmal ein dickes Lob an Stefan Becker, den Autor der Dokumentationshilfe in der 64'er, Ausgabe 6/85, für sein hervorragendes Programm. Aber auch hier gilt: Kein Programm, das nicht noch besser werden könnte.

Als erstes fiel mir auf, daß eine Dokumentation von Simons-Basic-Programmen nicht möglich war. Hier die vorläufige Lösung: Folgende Zeilen müssen eingefügt werden:

```
495 if x=100 then 330
805 if x=34 then y=2
806 if x=100 then y=3
835 if y=3 and (x=49 or x=50 or x=51) then y=0
836 if y=3 goto 750
```

Folgende Zeilen müssen geändert werden:

```
770 if x=34 or x=100 or x=131 or x=143 then z=0:
    goto 800
820 if x=34 and y=2 goto 750
920 ...:va$(i)=v1$+" :"
```

Des weiteren empfand ich es als einigermaßen lästig, bei vielen Variablen ständig die Frage »Wollen Sie Bemerkungen eingeben?« beantworten zu müssen. Nach folgenden Änderungen kann man die Variablen beliebig durchsuchen und mit Text versehen, um danach durch Eingabe von »E« wie Ende zur Ausgabe zu gelangen. Außerdem kann man nun bei Variablen, die sich über mehrere Druckzeilen erstrecken, seine Bemerkungen auf alle zugehörigen Zeilen verteilen. Folgende Zeilen sind einzufügen:

```
1105 if a$ ="e" goto 1180
1135 print "{home}{9down}"le$
```

Folgende Zeilen müssen geändert werden:

```
1080 ..a$< >chr$(13) and a$< >"e" goto 1080
1020 open 1,0
1160 ...: goto 1070
```

Die folgenden Zeilen sind zu löschen:

```
1140
1150
1170
```

Bei der Ausgabe ohne Zeilennummer genügt es im allgemeinen, wenn eine Variable, auch wenn sie sich über mehrere Zeilen erstreckt, nur einmal ausgegeben wird. Die folgende Lösung berücksichtigt dabei die Bemerkungen, die sich über mehrere Zeilen erstrecken. Es sind folgende Zeilen zu ändern:

```
1272 ...:goto 1275
1273 if left$(va$(i+1),4)=left$(va$(i),4) then 1275
1274 print#1,left$(va$(i), 8)
1275 next i
```

Die Abfrage »Nochmal/Weiter« läßt sich bei einer Ausgabe auf den Drucker durch Einfügen folgender Zeilen vermeiden:

```
685 if dv=4 then 712
1245 if dv=4 then 1269
```

Meine letzte Anmerkung betrifft die leidige Garbage-Collection (siehe hierzu auch den Kurs »Effektives Programmieren«, 64'er, Ausgabe 1/85), die der Computer durchführt, wenn sein Stringspeicher voll ist, und er weiteren Platz braucht. Bei meinen längeren Programmen lag der Computer spätestens bei der Eingabe von Bemerkungen für längere Zeit stumm und schweigend vor mir. Folgende Änderungen können hier Verbesserung schaffen:

1) Die Variable OB in Zeile 150 verkleinern. Sie gibt die Größe der Stringfelder an. Die Dauer der Garbage-Collection hängt aber in etwa vom Quadrat der Variablenzahl ab. Ich halte eine Angabe von OB=200 im Normalfall für völlig ausreichend. Ansonsten erscheint während des Programmlaufs eine entsprechende Fehlermeldung.

2) Das Ersetzen der speicherplatzverschwendenden Sortierprozedur durch die SWAP-Routine aus der 64'er, Ausgabe 1/85, Seite 123. Hierzu

1) Zeile 550 ersetzen durch:

```
550 sys 6000(ps$(i), ps$(j))
```

2) Zeile 1000 ersetzen durch:

```
1000 sys 6000(va$(i),va$(j))
```

3) Eingabe im Direktmodus:

```
print (peek(45)+256*peek(46))
```

Den erhaltenen Wert aufschreiben!

4) In den Zeilen 550 und 1000 den Wert 6000 durch den aufgeschriebenen Wert ersetzen.

5) Das Programm mit SAVE"XREF1.0",8 abspeichern.

6) Den Basic-Lader der SWAP-Routine laden und starten.

7) Auf die Frage nach der Startadresse den aufgeschriebenen Wert eingeben.

8) Das Dokumentationsprogramm mit LOAD"XREF1.0",8 laden.

9) Im Direktmodus eingeben (für »W« dabei den aufgeschriebenen Wert einsetzen):

```
POKE 45,(W+50)AND255:POKE 46,(W+50)/256
POKE174,(W+50)AND255:POKE175,(W+50)/256
```

Das Programm kann nun mit SAVE"XREF2.0",8 abgespeichert werden.

Achtung! Es dürfen keine Veränderungen mehr vorgenommen werden, die die Länge des Programms beeinflussen, da das Maschinenprogramm direkt hinter dem Basic-Programm steht! Das Zwischenprodukt "XREF1.0" kann jetzt gelöscht werden.

(Lutz Kleinostendarp/tr)

Anpassung von Cross-Ref auf Datasette:

Bedingt durch ein anderes Speicherformat auf Datasette gegenüber der Floppy ist Cross-Ref mit den in 64'er, Ausgabe 6/86 genannten Änderungen für Kassettenbetrieb trotzdem nicht lauffähig. Auch die Abfrage der Status-Variablen ist nicht korrekt.

Um Cross-Ref trotzdem auch für unsere Datasettenbesitzer zugänglich zu machen, ist folgendes zu beachten:

Das zu bearbeitende Programm muß in einem »Pseudo-Floppy-Format« gespeichert werden. Dazu das Programm laden und mit folgenden Befehlen im Direktmodus (also ohne Zeilennummern) neu aufs Band schreiben.

```
OPEN 1,1,1,"name" [RETURN]
PRINT#1,CHR$(PEEK(43));CHR$(PEEK(44) [RETURN]
FORI=PEEK(43)+256*PEEK(44)TOPEEK(45)+256*PE-
EK(46):PRINT#1,CHR$(PEEK(I));:NEXT [RETURN]
CLOSE1 [RETURN]
```

Das File hat zwar jetzt das richtige Format, aber die Statusabfrage in Zeile 90 ergibt immer noch falsche Werte. Auf Diskette zeigt ein ST=64 nämlich das Ende eines Files an, auf Kassette dagegen bedeutet ST=64, daß ein CHR$(0) folgt! Die folgenden Änderungen berücksichtigen dies:

```
90 GET#1,A$ : X=ASC(A$+N$) : IF X < >0 THEN
XX=0 : RETURN
95 XX=XX+1 : IF XX<3 THEN RETURN
290 OPEN 1,1,0,NA$ : GOSUB 120 : IF X=2049 THEN
310
730 OPEN 1,1,0,NA$ : GOSUB 120
```

Die Zeilen 1290 bis 1310 entfallen.

(Stefan Becker/tr)

Anmerkung der Redaktion:

Auf der Leserservice-Diskette dieser Ausgabe befindet sich »Cross-Ref« in der verbesserten Version von Gerhard Löber und die Simons Basic-Version von Lutz Kleinostendarp. Auch ist die Löber-Version zusätzlich compiliert worden. Diese Version können wir allen Lesern wegen der gewaltigen Geschwindigkeitsvorteile nur empfehlen.



Programm-Generator für den C 64

Diese Routine kann Basic-Zeilen innerhalb eines Basic-Programms erzeugen, löschen und in einen String einlesen, ohne Variablen zu zerstören.

Das Programm (Listing 1) ist vollständig in Maschinensprache geschrieben und belegt den Speicherbereich von $C800 bis $CA70. Nach dem Laden ist eine Initialisierung nicht notwendig. Um die Basic-Zeiger zurückzusetzen, ist nur ein NEW <RETURN> einzugeben.

Zum Erzeugen einer Basic-Zeile ist die Zeile selbst in einem String abzulegen. Anschließend kann dieser String mit
SYS 51200, String
in das Basic-Programm eingebunden werden. Zum Beispiel:
A$="100 S=SQR (2):PRINT S,S+100":SYS 51200,A$

Mit dem Befehl SYS 51442, Zeilennummer, läßt sich eine Basic-Zeile innerhalb eines Basic-Programms löschen.

Der Befehl
SYS 51442,100
löscht zum Beispiel die Zeile 100. Beim Generieren oder Löschen ergeben sich folgende Einschränkungen:
1. Der Aufruf (SYS 51200,A$ oder SYS 51442,100) darf nicht innerhalb einer FOR-NEXT-Schleife stehen.
2. SYS 51200,A$ oder SYS 51442,100 darf nicht in einem Unterprogramm stehen, falls sich die neue oder die zu löschende Zeile vor diesem Unterprogramm befindet.

Auch kann der Programm-Generator Zeilen in einen String einlesen. Dazu muß der String vorher jedoch mit beliebigen Zeichen gefüllt werden. Dabei gilt: Anzahl der Zeichen > = Anzahl der Zeichen in der zu erwartenden Zeile.

Es bietet sich natürlich an, den String mit Leerzeichen (CHR$(32)) zu belegen.

Anschließend kann die Zeile mit
SYS 51642, String, Zeilennummer eingelesen werden.
Beispiel:

```
100  S=SQR(2):PRINT S,S+100
110  A$="
120  SYS 51642,A$,100
130  PRINT A$
```

Nach dem Start dieses Beispiels mit RUN <RETURN> erscheint auf dem Bildschirm die Zeile 100:

S=SQR(2): PRINT S, S+100

READY.

Für dieses Programm gibt es nun eine Anzahl von Anwendungsmöglichkeiten. Es läßt sich zum Beispiel ein nur acht Zeilen langes Basic-Programm schreiben, das ein Maschinenprogramm in DATA-Zeilen ablegt (Listing 2).

Dabei wird etwa die Geschwindigkeit erreicht, die die entsprechende Routine im SMON hat. Außerdem wäre ein »intelligentes« Programm vorstellbar, das sich selbst verändert. Typische Maschinenroutinen wie RENUMBER, FIND und AUTO sind natürlich auch denkbar. Sie sehen, die Möglichkeiten dieses Programm-Generators sind nahezu unbegrenzt.

(Michael Dietz/ah)

```
10 Z=100:A=10:I=51200:E=51810                          <148>
20 A$=STR$(Z)+"DATA ":Z=Z+A                            <183>
30 S$=STR$(PEEK(I)):A$=A$+RIGHT$(S$,LEN(S$
   )-1)+","                                            <049>
40 IF LEN(A$)<35 THEN I=I+1:IF I<E THEN 30             <040>
50 A$=LEFT$(A$,LEN(A$)-1)                              <076>
60 SYS 51200,A$:PRINT A$:I=I+1:IF I<E THEN
   20                                                  <073>
70 I=10:GOTO 50000                                     <068>
50000 SYS 51442,I:I=I+10:IF I<71 THEN 5000
      0                                                <116>

© 64'er
```

Listing 2. Mit nur acht Basic-Zeilen lassen sich Maschinenprogramme in DATA-Zeilen ablegen.

```
programm : 'programm gen      c800 ca62
----------------------------------------

c800 : 20 fd ae 20 9e ad 20 a3   ee
c808 : b6 a0 00 85 57 b1 22 99   7e
c810 : 00 02 c8 c4 57 d0 f6 a9   07
c818 : 00 99 00 02 a5 7a 8d 70   6a
c820 : ca a5 7b 8d 71 ca a9 ff   61
c828 : 85 7a a9 01 85 7b 20 73   10
c830 : 00 20 6b a9 20 79 a5 84   be
c838 : 0b 8c 72 ca 20 13 a6 18   e5
c840 : a5 31 85 57 6d 72 ca 85   6b
c848 : 59 a5 32 85 58 69 00 85   8d
c850 : 5a 38 a5 5f e9 01 8d 73   df
c858 : ca a5 60 e9 00 8d 74 ca   1e
c860 : a0 00 b1 57 91 59 38 a5   68
c868 : 57 e9 01 85 57 a5 58 e9   7d
c870 : 00 85 58 38 a5 59 e9 01   1f
c878 : 85 59 a5 5a e9 00 85 5a   c8
c880 : ad 74 ca c5 58 d0 db ad   a9
c888 : 73 ca c5 57 d0 d4 a2 00   fb
c890 : 18 b5 2d 6d 72 ca 95 2d   aa
c898 : e8 b5 2d 69 00 95 2d e8   07
c8a0 : e0 08 d0 ec a0 00 a5 5f   b5
c8a8 : 91 5f c8 a5 60 91 5f c8   71
c8b0 : a5 14 91 5f c8 a5 15 91   e1
c8b8 : 5f c8 b9 fc 01 91 5f c8   35
c8c0 : cc 72 ca d0 f5 a0 00 18   27
c8c8 : b1 5f 6d 72 ca 91 5f 8d   a4
c8d0 : 73 ca c8 b1 5f 69 00 91   75
c8d8 : 5f 8d 74 ca ad 73 ca 85   21
c8e0 : 5f ad 74 ca 85 60 b1 5f   6d
c8e8 : d0 db a9 00 8d 73 ca 4c   48
c8f0 : 79 c9 20 fd ae 20 8a ad   87
c8f8 : 20 f7 b7 a5 7a 8d 70 ca   22
c900 : a5 7b 8d 71 ca 20 13 a6   3c
c908 : b0 01 60 a0 00 38 b1 5f   ac
c910 : 85 57 e5 5f 8d 72 ca c8   cf
c918 : b1 5f 85 58 88 a5 5f 85   23
c920 : 59 a5 60 85 5a b1 57 91   c8
c928 : 59 18 a5 57 69 01 85 57   45
c930 : a5 58 69 00 85 58 18 a5   22
c938 : 59 69 01 85 59 a5 5a 69   36
c940 : 00 85 5a a5 32 c5 58 d0   a2
c948 : dc a5 31 c5 57 d0 d6 a0   94
c950 : 00 38 b1 5f ed 72 ca 91   85
c958 : 5f 8d 73 ca c8 b1 5f e9   1f
c960 : 00 91 5f 8d 74 ca ad 73   ee
c968 : ca 85 5f ad 74 ca 85 60   f7
c970 : b1 5f d0 db a9 ff 8d 73   38
c978 : ca a5 9d d0 32 38 a5 14   3a
c980 : e5 39 a5 15 e5 3a b0 27   4f
c988 : ad 73 ca d0 11 18 ad 70   25
c990 : ca 6d 72 ca 85 7a ad 71   cc
c998 : ca 69 00 85 7b 60 38 ad   bf
c9a0 : 70 ca ed 72 ca 85 7a ad   5d
c9a8 : 71 ca e9 00 85 7b 60 ad   0a
c9b0 : 70 ca 85 7a ad 71 ca 85   d3
c9b8 : 7b 60 20 fd ae 20 9e ad   ed
c9c0 : 20 a3 b6 a5 22 85 59 a5   13
c9c8 : 23 85 5a 20 fd ae 20 9e   5b
c9d0 : ad 20 f7 b7 20 13 a6 b0   19
c9d8 : 03 4c 61 ca a5 14 85 63   8b
c9e0 : a5 15 85 62 a2 90 38 20   8d
c9e8 : 49 bc 20 df bd a0 ff c8   06
c9f0 : b9 00 01 91 59 d0 f8 a9   6f
c9f8 : 20 91 59 c8 84 57 a0 03   dc
ca00 : 84 58 e6 58 a4 58 b1 5f   08
ca08 : f0 57 c9 22 d0 19 a4 57   72
ca10 : 91 59 e6 57 e6 58 a4 58   67
ca18 : b1 5f c9 22 f0 0d a4 57   e8
ca20 : 91 59 e6 57 4c 14 ca c9   26
ca28 : 80 10 09 a4 57 91 59 e6   bc
ca30 : 57 4c 02 ca c9 ff f0 f3   cf
ca38 : 38 e9 7f aa a0 ff ca f0   b1
ca40 : 08 c8 b9 9e a0 10 fa 30   c5
ca48 : f5 c8 b9 9e a0 30 0d 84   ac
ca50 : 5b a4 57 91 59 e6 57 a4   79
ca58 : 5b 4c 49 ca 29 7f 4c 2b   9b
ca60 : ca 60 00 00 00 00 00 00   5b
```

Listing 1 »Programm-Generator«. Bitte beachten Sie die Eingabehinweise auf Seite 54.

Bitte beachten Sie die Eingabehinweise auf Seite 54.

Neues vom SMON

Diese Erweiterung stellt elf weitere Befehle zur Verfügung. So läßt sich der Monitor zum Beispiel frei im Speicher verschieben und Sprites oder Zeichensätze können sehr einfach erstellt und geändert werden.

Um die Befehlserweiterung zu initialisieren, geht man folgendermaßen vor:
1. SMON absolut laden.
2. Den Basic-Lader (siehe Listing) eintippen und abspeichern.
3. Nach dem Start des Laders die Startadresse (dezimal) Ihrer SMON-Version eingeben:
zum Beispiel 49152 (= $C000).
4. Den erweiterten SMON zum Beispiel mit "SMONEX" Startadresse Endadresse abspeichern.

Die neuen Routinen werden, genau wie die meisten bereits vorhandenen, durch einen Buchstaben, zum Teil gefolgt von Adressenangaben, aufgerufen. Bei den ersten drei Ausgabebefehlen kann der Speicherinhalt durch Überschreiben der Zeile geändert werden.

Z 4000 (4100) (Zeichendaten)
gibt den Speicherinhalt von $4000 (bis $40FF) folgendermaßen aus: Jeweils ein Byte pro Zeile wird in 8-Bit-Form dargestellt. Dabei ist ein »*« ein gesetztes, ein ».« dagegen ein nicht gesetztes Bit. Die beiden Zeichen sind willkürlich gewählt und können durch Überschreiben der Speicherzellen $xE65, $xE2D (Bit = 1) und $xE69, $xE30 (Bit = 0) in den Bildschirm-Code (!) der gewünschten Zeichen geändert werden.

Die Anwendung dieses Befehls liegt beispielsweise in der gezielten und anschaulichen Beeinflussung bestimmter Steuerbits in VIC, CIA, etc. Andererseits lassen sich — besonders in Verbindung mit dem Kommando »Q« — Zeichendaten leicht modifizieren.

H 4000 (4100)
entspricht dem Befehl »Z« mit dem Unterschied, daß jeweils drei Byte pro Zeile ausgegeben werden. Das entspricht dem Format für Spritedaten. Auf diese Weise steht mit dem erweiterten SMON ein kleiner »Sprite-Editor« zur Verfügung.

N 4000 (4100) (Normaldarstellung)
interpretiert den Speicherinhalt von $4000 (bis $40FF) als Bildschirm-Code und gibt 32 Zeichen pro Zeile aus.

U 4000 (4100) (Übersicht)
Wie »N«, jedoch werden in einer Zeile 40 Zeichen dargestellt. Änderungen sind nur mit »N« möglich. Dieser Befehl dient hauptsächlich dazu, im Speicher abgelegte Bildschirminformationen so auszugeben, wie sie tatsächlich im 40-Zeichen/Zeile-Format aussehen würden. Dieser Befehl ist recht nützlich, um professionelle Videospiele zu analysieren, da hier Spielszenen oft im Bildschirm-Code gespeichert sind.

E 4000 (4100) (Erase)
ist der bereits in 64'er Ausgabe 2/85 vorgeschlagene Erase-Befehl zum Füllen des Speicherbereiches von $4000 bis $40FF mit $00.

Y 40
kopiert die vorhandene SMON-Version in nur drei Sekunden nach $4000 bis $4FFF und nimmt dabei alle notwendigen Anpassungen vor. Die ursprüngliche Speicherversion des Monitors bleibt unverändert. Mit »G 4000« kann man in den neuen SMON springen. Von dem Byte-Wert, der übergeben werden muß, wird nur das obere Nibble ($4) gewertet, so daß sich theoretisch 16 SMON-Versionen im Speicher unterbringen lassen, wobei natürlich nicht alle Möglichkeiten sinnvoll sind.

Auf diese Weise läßt sich stets die erforderliche Speicherversion herstellen, ohne daß langwierige Änderungen notwendig sind.

Q 2000
kopiert den Zeichensatz aus dem ROM von $D000 bis $DFFF in das RAM nach $2000. Dort kann er mit dem Befehl »Z« nach Belieben geändert werden. Möchte man zum Beispiel das Zeichen »A« in ein »Ä« umdefinieren, so ist der Zeichensatz mit »Q 2000« ins RAM zu kopieren. Anschließend kann mit »Z 2000 2015« der Bereich in binärer Form auf dem Bildschirm ausgegeben werden, in dem auch das Zeichen »A« steht. Dieses kann nun (siehe Bild) in ein »Ä« geändert werden, indem man mit dem Cursor an die zu ändernde Stelle fährt und für einen Punkt, der gesetzt werden soll, ein »*« und für einen Punkt der nicht gesetzt werden soll ein ».« setzt. So, jetzt ist der Zeichensatz umdefiniert, aber noch nicht aktiviert. Als nächstes muß dem Videocontroller die Startadrese des neuen Zeichensatzes mitgeteilt werden. Dazu ist die Adresse $D018, in der eine hexadezimale 15 steht, durch eine hexadezimale 18 zu ersetzen.

J
bringt den letzten Ausgabebefehl (K, D, M, Z, H, N, U) auf den Bildschirm zurück. Mit RETURN wird der letzte Befehl noch einmal ausgeführt.

Zum Schluß noch ein Tip:
DATA-Zeilen in Hex-Byte-Darstellung sind wegen ihrer konstanten Länge (immer zwei Ziffern pro Wert!) übersichtlicher als solche mit dezimalen Zahlen. Da für die Ausgabe von Hex-Werten bereits alle Routinen im SMON integriert sind, kann der »B«-Befehl (Basic-DATA-Zeilen erzeugen) durch Verändern eines einzigen Sprungbefehles dahingehend manipuliert werden, daß der Speicherinhalt künftig in Form von Hex-Byte ausgegeben wird:

Disassemblieren Sie dazu den Byte-Ausgabebefehl mit »D x99F« und ersetzen »JSR BDD1« durch »JSR x32A«. Für das »x« muß der 4-KByte-Block, in dem die zu ändernde SMON-Version steht, eingesetzt werden. Liegt Ihre SMON-Version bei $C000, so ersetzen Sie das »x« durch ein »C«.

Die Gesamtlänge der DATA-Zeile kann außerdem durch Verändern der Speicherzelle $x9AE variiert werden. Bei dem Wert $1C werden zum Beispiel genau acht Hex-Byte pro Zeile ausgegeben. (Mark Richters/ah)

```
100 REM ******************************    <238>
110 REM *                            *    <159>
120 REM *      SMON - ERWEITERUNG    *    <222>
130 REM *                            *    <179>
140 REM *  VON   MARK RICHTERS       *    <005>
150 REM *        ALLERSTR.4          *    <068>
160 REM *        2806 OYTEN          *    <037>
170 REM *        TEL.: 04207/1870    *    <078>
180 REM *                            *    <229>
190 REM ******************************    <072>
200 :                                     <176>
210 PRINT"BITTE DIE STARTADRESSE"         <248>
220 PRINT"IHRER SMON-VERSION EINGEBEN!"   <189>
230 INPUT H : H=H/4096                    <187>
240 IF H<>INT(H) THEN 210                 <052>
```

Erweiterung zum SMON. Bitte die Eingabe-Hinweise auf Seite 54 beachten.

```
250 :                                   <226>
260 :                                   <238>
270 DIM W(75)                           <045>
280 FOR I=0 TO 9                        <096>
290 :  W(48+I)=I                        <224>
300 :  W(65+I)=I+10                     <069>
310 NEXT I                              <140>
320 :                                   <042>
330 FOR I=1 TO 4                        <048>
340 :  READ K,Z : K=K+H*4096            <123>
350 :  FOR J=K TO K+Z-1                 <235>
360 :    READ A$                        <038>
370 :    A=W(ASC(A$))                   <253>
380 :    B=W(ASC(RIGHT$(A$,1)))         <129>
390 :    S=S+A+B                        <023>
400 :    IF ASC(A$)=42 THEN A=H         <010>
410 :    P=A*16+B : POKE J,P            <118>
420 :  NEXT J                           <070>
430 NEXT I                              <004>
440 :                                   <162>
450 IF S=7031 THEN PRINT"OK!":GOTO 470  <200>
460 PRINT"FEHLER IN DATAS !"            <071>
470 STOP                                <026>
480 :                                   <202>
490 DATA 97,7                           <136>
500 DATA 27,3A,3B,2C,28,29,21           <025>
510 :                                   <232>
520 DATA 739,28                         <137>
530 DATA A2,06,DD,61,*0,F0,08,CA        <159>
540 DATA 10,F8,A9,2E,20,D2,FF,20        <187>
550 DATA CA,*2,C9,2E,F0,F9,4C,85        <184>
560 DATA*F,4C,B2,*F                     <109>
570 :                                   <038>
580 DATA 781,2                          <220>
590 DATA F0,ED                          <141>
600 :                                   <068>
610 DATA 3593,503                       <202>
620 DATA A9,80,2C,A9,00,85,AB,20        <085>
630 DATA 64,*2,24,AB,10,03,A2,29        <117>
640 DATA 2C,A2,28,20,40,*3,20,23        <239>
650 DATA*3,A0,06,A2,00,A1,FB,0A         <071>
660 DATA 48,B0,03,A9,2E,2C,A9,2A        <144>
670 DATA 91,D1,AD,86,02,91,F3,68        <079>
680 DATA C8,E8,E0,08,D0,E9,20,67        <139>
690 DATA*3,24,AB,30,04,C0,1E,90         <233>
700 DATA DA,20,5D,*4,90,C4,60,A0        <133>
710 DATA 08,2C,A0,18,20,7E,*2,20        <100>
720 DATA B8,*2,A2,08,A9,00,85,AA        <236>
730 DATA 20,CA,*2,C9,2E,F0,07,C9        <052>
740 DATA 2A,F0,04,4C,D1,*2,18,26        <202>
750 DATA AA,88,CA,D0,EB,A5,AA,81        <014>
760 DATA FB,C1,FB,D0,EE,20,67,*3        <071>
770 DATA C0,00,D0,D6,60,A9,80,2C        <218>
780 DATA A9,00,85,AB,20,64,*2,20        <243>
790 DATA 51,*3,24,AB,10,0B,A9,21        <138>
800 DATA 20,D2,FF,20,23,*3,A0,08        <033>
810 DATA 2C,A0,00,A2,00,A1,FB,20        <242>
820 DATA 4F,*4,D0,F9,20,5D,*4,90        <059>
830 DATA DE,60,20,7E,*2,A2,00,A0        <137>
840 DATA 08,B1,D1,81,FB,C1,FB,D0        <193>
850 DATA AA,20,56,*4,90,F3,60,20        <193>
860 DATA 8D,*2,29,F0,85,FF,20,3C        <190>
870 DATA*F,20,D6,*9,20,3C,*F,A9         <001>
880 DATA 14,85,FB,A9,02,05,FF,85        <195>
890 DATA FC,A9,D1,85,FD,A9,0F,05        <008>
900 DATA FF,85,FE,20,68,*A,A0,00        <182>
910 DATA A2,0D,BD,F2,*F,05,FF,85        <072>
920 DATA FC,CA,BD,F2,*F,85,FB,B1        <238>
930 DATA FB,29,0F,05,FF,91,FB,CA        <150>
940 DATA 10,E8,A9,2B,85,FB,A9,00        <089>
950 DATA 05,FF,85,FC,A0,35,B1,FB        <082>
960 DATA 29,0F,05,FF,91,FB,88,88        <137>
970 DATA 10,F4,A9,DE,85,FB,A9,0F        <098>
980 DATA 05,FF,85,FC,A0,13,B1,FB        <088>
990 DATA 29,0F,05,FF,91,FB,88,88        <167>
1000 DATA 10,F4,60,A5,FF,85,A9,20       <096>
1010 DATA 43,*F,68,68,29,F0,85,A5       <117>
1020 DATA 18,69,10,85,A7,A9,00,85       <054>
1030 DATA A4,85,A6,85,A8,60,20,7A       <029>
1040 DATA*2,A9,00,4C,C7,*9,20,7E        <250>
1050 DATA*2,A0,00,A9,D0,84,FD,85        <002>
1060 DATA FE,78,A9,03,85,01,A2,10       <164>
1070 DATA B1,FD,91,FB,C8,D0,F9,E6       <125>
1080 DATA FC,E6,FE,CA,D0,F2,A9,27       <124>
1090 DATA 85,01,58,60,48,C9,4A,D0       <178>
1100 DATA 10,A0,27,B9,00,02,91,D1       <224>
1110 DATA 88,10,F8,68,C6,D6,4C,D6       <214>
1120 DATA*2,A0,06,D9,D7,*F,D0,0A        <187>
1130 DATA A0,27,B1,D1,99,00,02,88       <003>
1140 DATA 10,F8,88,10,EE,68,4C,FF       <033>
1150 DATA*2,A2,0A,DD,D0,*F,F0,06        <139>
1160 DATA CA,D0,F8,4C,D1,*2,20,C5       <228>
1170 DATA*F,4C,D6,*2,8A,0A,AA,BD        <009>
1180 DATA DD,*F,48,BD,DC,*F,48,60       <111>
1190 DATA 28,29,21,45,59,51,48,5A       <209>
1200 DATA 4E,55,44,4B,4D,52,*E,4F       <203>
1210 DATA*E,B2,*E,56,*F,C7,*E,5E        <099>
1220 DATA*F,0B,*E,08,*E,85,*E,88        <157>
1230 DATA*E,06,00,87,03,2D,0C,5C        <076>
1240 DATA 0C,F5,0C,A2,0D,04,0E          <224>
```


Erweiterung zum SMON. (Schluß)

```
)4000 ..****..  )4000 ..****..
)4001 .**..**.  )4001 .**..**.
)4002 .**.***.  )4002 .**.***.
)4003 .**.***.  )4003 .**.***.
)4004 .**.....  )4004 .**.....
)4005 .**...*.  )4005 .**...*.
)4006 ..****..  )4006 ..****..
)4007 ........  )4007 ........
)4008 ...**...  )4008 .**..**.
)4009 ..****..  )4009 ........
)400A .**..**.  )400A ..****..
)400B .******.  )400B .**..**.
)400C .**..**.  )400C .******.
)400D .**..**.  )400D .**..**.
)400E .**..**.  )400E .**..**.
)400F ........  )400F ........
)4010 .*****..  )4010 .*****..
)4011 .**..**.  )4011 .**..**.
)4012 .**..**.  )4012 .**..**.
)4013 .*****..  )4013 .*****..
)4014 .**..**.  )4014 .**..**.
```

Bild. Originaler und abgeänderter Zeichensatz. Aus dem A wird ein Ä.

Programme nachladen

Wenn man von Basic aus ein Programm nachladen will, so ist im allgemeinen die Tatsache störend, daß der Interpreter anschließend das Basic-Programm neu startet. Mit folgender Befehlsfolge wird dies umgangen:

SYS 57812"name",8,1:POKE 780,0:SYS 65493

Das gewünschte Maschinenprogramm wird geladen, und das Basic-Programm mit dem nächsten Befehl fortgesetzt.

Wenn man statt ...,8,1 die Sekundäradresse ...,8,0 setzt, und im X- und Y-Register des 6510 (Speicherstellen 781 und 782) Low-Byte und High-Byte einer bestimmten Adresse angibt, so wird das Programm an diese Adresse geladen. Eine Anwendung wäre zum Beispiel eine Merge-Routine:

```
1000 AD=PEEK(45)+PEEK(46)*256—2
1010 AH=INT(AD/256)
1020 AL=AD—256*AH
1030 POKE 781,AL
1040 POKE 782,AH
1050 SYS 57812"name",8,0
1060 POKE 780,0
1070 SYS 65493
```

(Martin Barth/tr)

Gelöschte Files retten

Hat man versehentlich ein Programm auf Diskette gelöscht, so gibt es eine einfache Möglichkeit, dieses wieder zu retten: LOAD "*",8. Voraussetzung ist allerdings, daß man inzwischen noch kein anderes File von Diskette geladen hat.

Dieser Befehl lädt das zuletzt bearbeitete Programm von Diskette; auch wenn dieses inzwischen gelöscht wurde!

(Michael Keukert/tr)

»Kopierschutz« für Datasette

Auf Datasette läßt sich sehr leicht ein wirkungsvoller Softwareschutz erzeugen:

Es besteht nämlich beim C 64 die Möglichkeit, einen bis zu 172 (!) Zeichen langen Programmnamen abzuspeichern. Dieser wird zwar beim Laden nicht angezeigt, steht aber trotzdem ab Adresse 849 im Speicher. Dort kann man ihn dann gezielt auf bestimmte Zeichen untersuchen.

(Peter Dreuw/tr)

Neue Farben für den SMON

Da die Farbkombination des SMON meiner Meinung nach recht ungünstig gewählt ist, hier mein Verbesserungs-Vorschlag (für SMON $C000):

Monitor Hintergrund und Rahmenfarbe hellblau	($C220: 0E)
	($CFC7: 0E)
Monitor Schriftfarbe schwarz	($C228: 00)
Basic: Schrift- und Rahmenfarbe blau	($C36F: 06)
Basic: Hintergrundfarbe hellblau	($C377: 0E)
Diskmonitor Rahmenfarbe blau	($CE0A: 06)

(Björn Hendrik Gemein/tr)

Hi-EDDI auf Star SG-10

Betrifft: Anpassung der Druckerroutine HI-PRINT von HI-EDDI an den SG-10.

Folgende Änderungen müssen an der Druckerroutine vorgenommen werden:

```
460 DATA 4: REM Direktkanal
550 DATA 27,51,15,255,255: REM Zeilenabstand 15/144 Inch
580 DATA 27,103,4,255,255: REM Grafikmodus 4
```

In dieser Betriebsart werden Kreise in Y-Richtung etwas gedehnt. Wer auf den Ausdruck zweier Bilder nebeneinander und auf Großbilder verzichtet, sollte folgende Werte eingeben (Zeile 460 muß natürlich trotzdem geändert werden):

```
550 DATA 27,51,16,255,255: REM Zeilenabstand 16/144 Inch
580 DATA 27,103,5,255,255: REM Grafikmodus 5
```

Dafür werden Kreise auch als Kreise dargestellt. Wenn man breiteres Papier als A4 verwendet, kann man auch zwei Bilder nebeneinander und Großbilder drucken. Der Drucker muß auf Star-Betriebsart eingestellt sein (siehe Handbuch). Der User-Port kann weiterhin als Parallelschnittstelle verwendet werden.

(Kurt Reiter/tr)

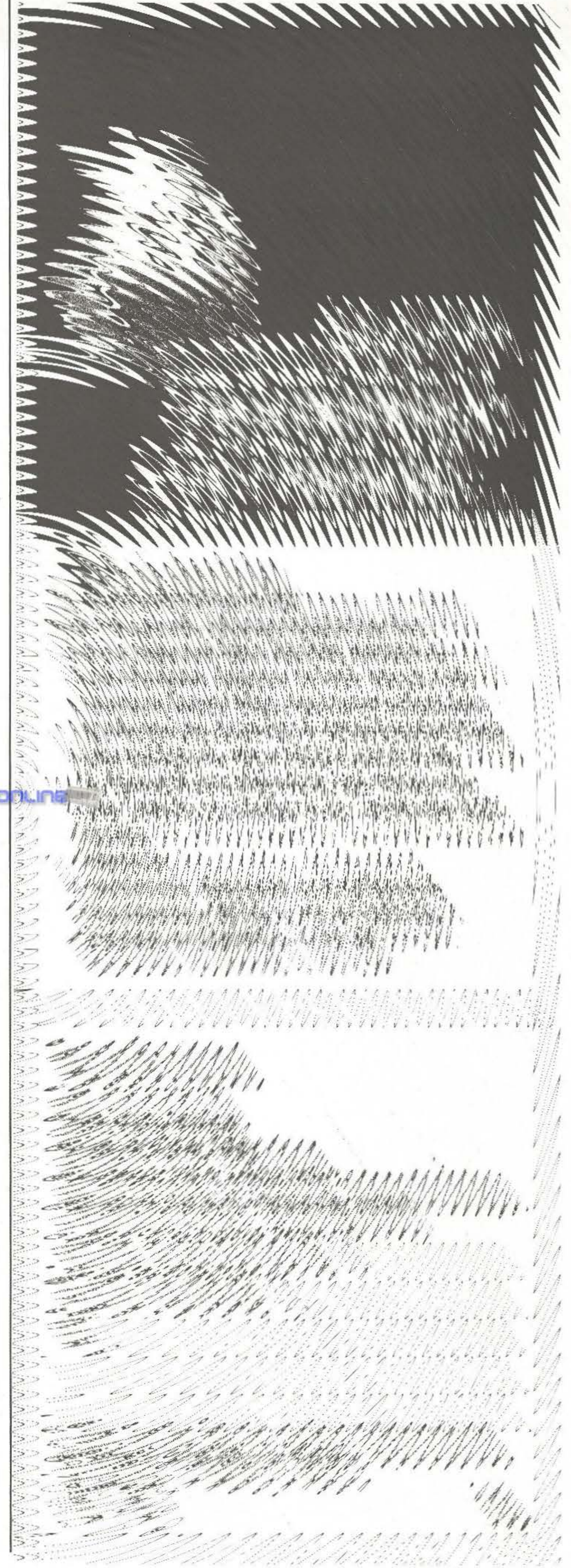
3D-Zeichensatz

Dreidimensionaler Zeichensatz auf dem C 64 (nicht SX 64)? Kein Problem, mittels eines einzigen POKE:

POKE 53270,250

Man sollte aber a) über einen Farbmonitor oder Farbfernseher verfügen, und b) die Hintergrundfarbe mit POKE 53281,4 auf violett setzen. Dadurch kommt der Effekt besser zur Geltung.

(Matthias Eberhardt/tr)

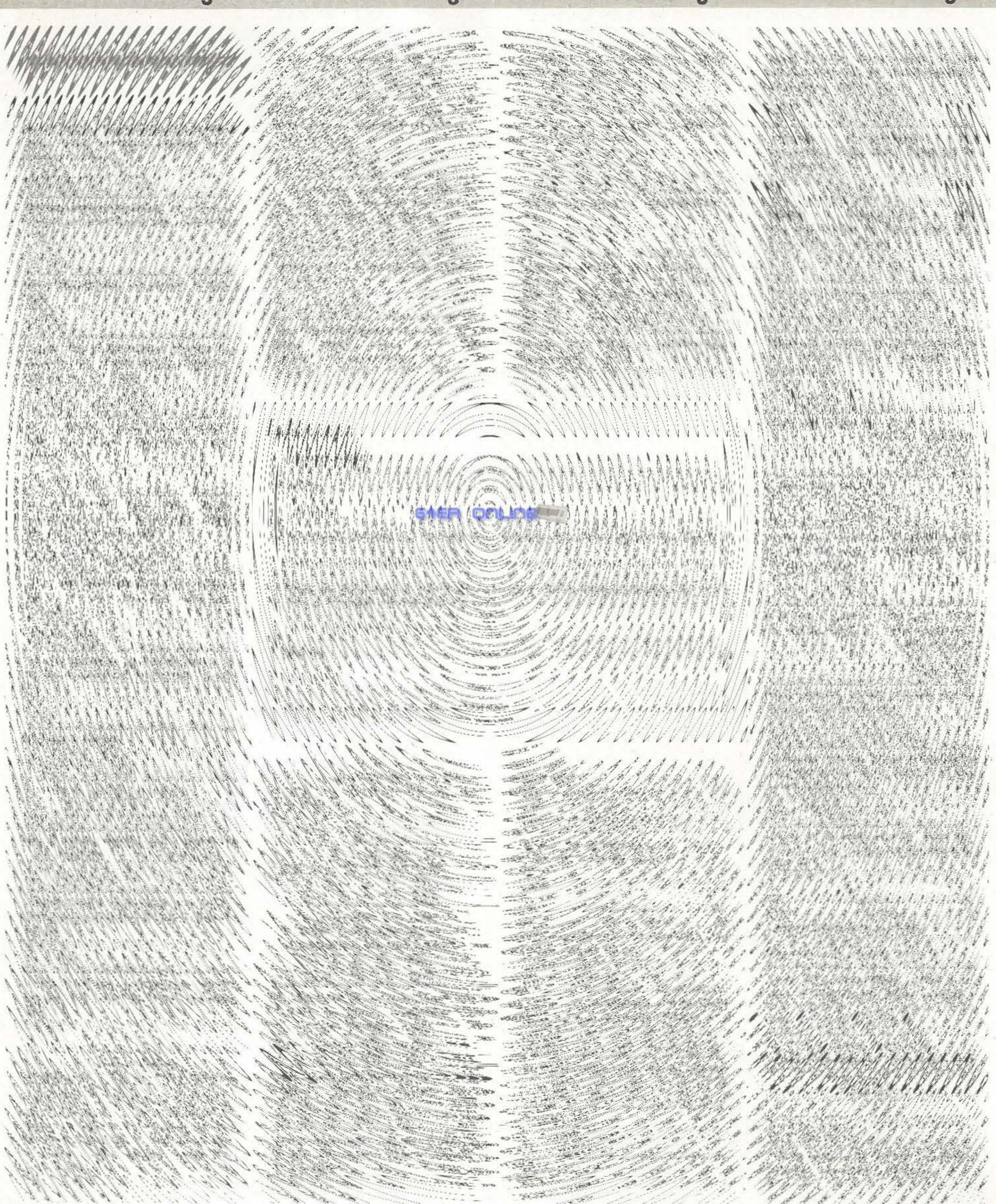


COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **November-Ausgabe** (erscheint am 18. Oktober 85): Schicken Sie Ihren Anzeigentext bis zum 25. September 85 (Eingangsdatum beim Verlag) an »64er«. Später eingehende Aufträge werden in der **Dezember-Ausgabe** (erscheint am 15. 11. 85) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes.
Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen.
Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen







Das 64'er Extra ist eine Extra-Leistung für unsere Leser. Sie finden hier geballte Informationen für Ihren C 64 zum Herausnehmen und Sammeln.

Im 64'er Extra 2 finden Sie eine komplette Übersicht über alle Register des Video-Chips im C 64. Damit haben Sie auf einem Blick alle Informationen über:
- Spritesteuerung
- Farbgebung
- hochauflösende Grafik
- Multicolor-Modus
- Zeichensatz
- Scrolling
- Rasterzeilen-Interrupt

Geballte Information, verständlich und übersichtlich präsentiert, das ist das Konzept des 64'er Extra.

Ob Sie sich die Extra-Seiten als Poster über Ihren Schreibtisch hängen, oder einmal in der Mitte gefaltet in einer Plastikhülle griffbereit neben Ihrem Computer unterbringen — in jedem Fall bekommen Sie mit der Zeit ein einzigartiges Archiv über Ihren Commodore 64 zusammen.

Diese zweite Ausgabe des 64'er Extra bringt Ihnen eine Übersicht über den VIC II-Chip, der in Ihrem C 64 alles steuert, was irgendwie mit Farbe, Grafik oder Zeichensatz zu tun hat. Zu jedem Register finden Sie die Bedeutung der einzelnen Bits im Klartext erläutert. Ob Anfänger oder Profi, hier haben Sie in jedem Fall eine kompakte, vollständige Übersicht über den VIC II. Sie ersparen sich damit umständliches Blättern im Handbuch und anderer Literatur, und brauchen sich auch nicht mit unverständlichen Bezeichnungen und Abkürzungen herumschlagen.

Bitpaar	Farbquelle	Register
00	Bildschirmfarbe	33
01	Sprite-Mehrfarbenregister 0	37
10	Sprite-Farbregister	39-46
11	Sprite-Mehrfarbenregister 1	38

Herkunft der vier Punktfarben im Sprite-Mehrfarben-Modus

Die Register des VIC II-Chips (msb = höchstwertiges Bit einer Adressen- oder Positionsangabe)

Register	Adresse dezimal	Adresse hex	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	53248	$ D000	X-Position des Sprite Nr. 0							
1	53249	$ D001	Y-Position des Sprite Nr. 0							
2	53250	$ D002	X-Position des Sprite Nr. 1							
3	53251	$ D003	Y-Position des Sprite Nr. 1							
4	53252	$ D004	X-Position des Sprite Nr. 2							
5	53253	$ D005	Y-Position des Sprite Nr. 2							
6	53254	$ D006	X-Position des Sprite Nr. 3							
7	53255	$ D007	Y-Position des Sprite Nr. 3							
8	53256	$ D008	X-Position des Sprite Nr. 4							
9	53257	$ D009	Y-Position des Sprite Nr. 4							
10	53258	$ D00A	X-Position des Sprite Nr. 5							
11	53259	$ D00B	Y-Position des Sprite Nr. 5							
12	53260	$ D00C	X-Position des Sprite Nr. 6							
13	53261	$ D00D	Y-Position des Sprite Nr. 6							
14	53262	$ D00E	X-Position des Sprite Nr. 7							
15	53263	$ D00F	Y-Position des Sprite Nr. 7							
16	53264	$ D010	Spr. 7, msb X-Pos.	Spr. 6, msb X-Pos.	Spr. 5, msb X-Pos.	Spr. 4, msb X-Pos.	Spr. 3, msb X-Pos.	Spr. 2, msb X-Pos.	Spr. 1, msb X-Pos.	Spr. 0, msb X-Pos.
17	53265	$ D011	msb des Rasterregisters (Reg. 18)	Schaltbit für veränderten Hintergrundfarbmodus 1 = eingeschaltet	Schaltbit für Hochauflösungsmodus 1 = eingeschaltet	Schaltbit für Bildschirm »aus« 0 = normaler Bildschirm 1 = Bildschirmfarbe	Schaltbit für Zeilenzahl 0 = 24 Zeilen 1 = 25 Zeilen	Wert der Zeilenverschiebung in Y-Richtung beim Smooth Scrolling		
18	53266	$ D012	Rasterregister. Dazu kommt das msb in Bit 7, Register 17							
19	53267	$ D013	Lichtgriffel X-Position							
20	53268	$ D014	Lichtgriffel Y-Position							
21	53269	$ D015	Ein- und Ausschalten von Sprites. 0 = Sprite aus, 1 = Sprite an Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
22	53270	$ D016	(unbenutzt)		Reset-Bit, muß 0 sein, damit VIC-II-Chip arbeitet	Schaltbit für Mehrfarbmodus 1 = eingeschaltet	Schaltbit für Spaltenzahl 0 = 38 Spalten 1 = 40 Spalten	Wert der Spaltenverschiebung in X-Richtung beim Smooth Scrolling		

Setzen einzelner Bits per Basic (ADR: Adresse eines Registers, BIT: gewünschtes Bit, also eine Zahl zwischen 0 und 7):
POKE ADR, PEEK (ADR) OR 2 ↑ BIT

Löschen einzelner Bits in VIC II-Registern per Basic:
POKE ADR, PEEK (ADR) AND (255 — 2 ↑ BIT)

Abfragen eines Bits:
? PEEK (ADR) AND 2 ↑ BIT

Es ergibt sich ein Wert von Null, wenn das Bit gelöscht ist und ein Wert ungleich Null, wenn das Bit gesetzt ist.
Beispiel: POKE 53277, PEEK (53277) OR 2 ↑ 7
bewirkt, daß Sprite 7 in X-Richtung vergrößert wird.

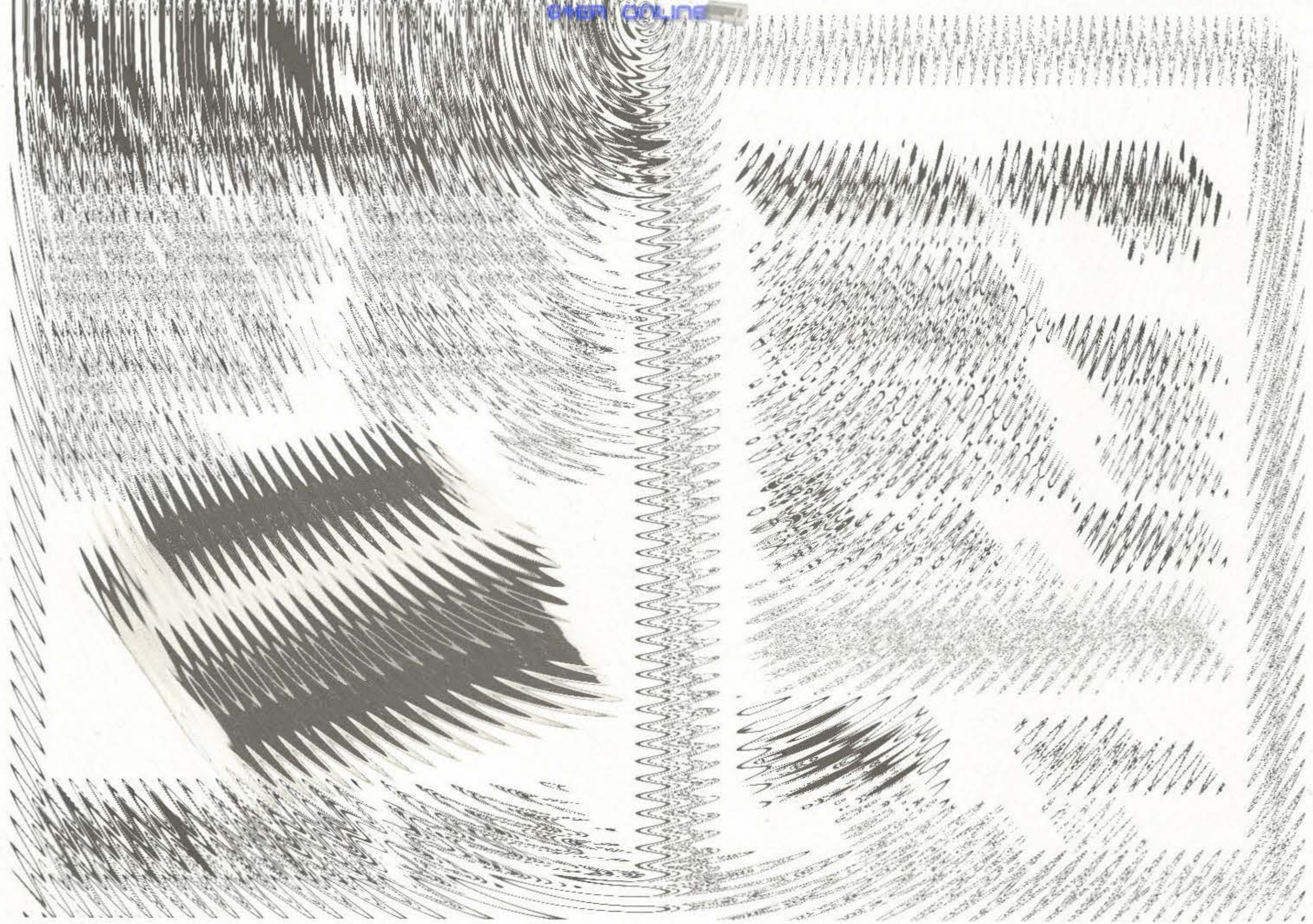
Ein Sprite-Entwurfsblatt zum fotokopieren

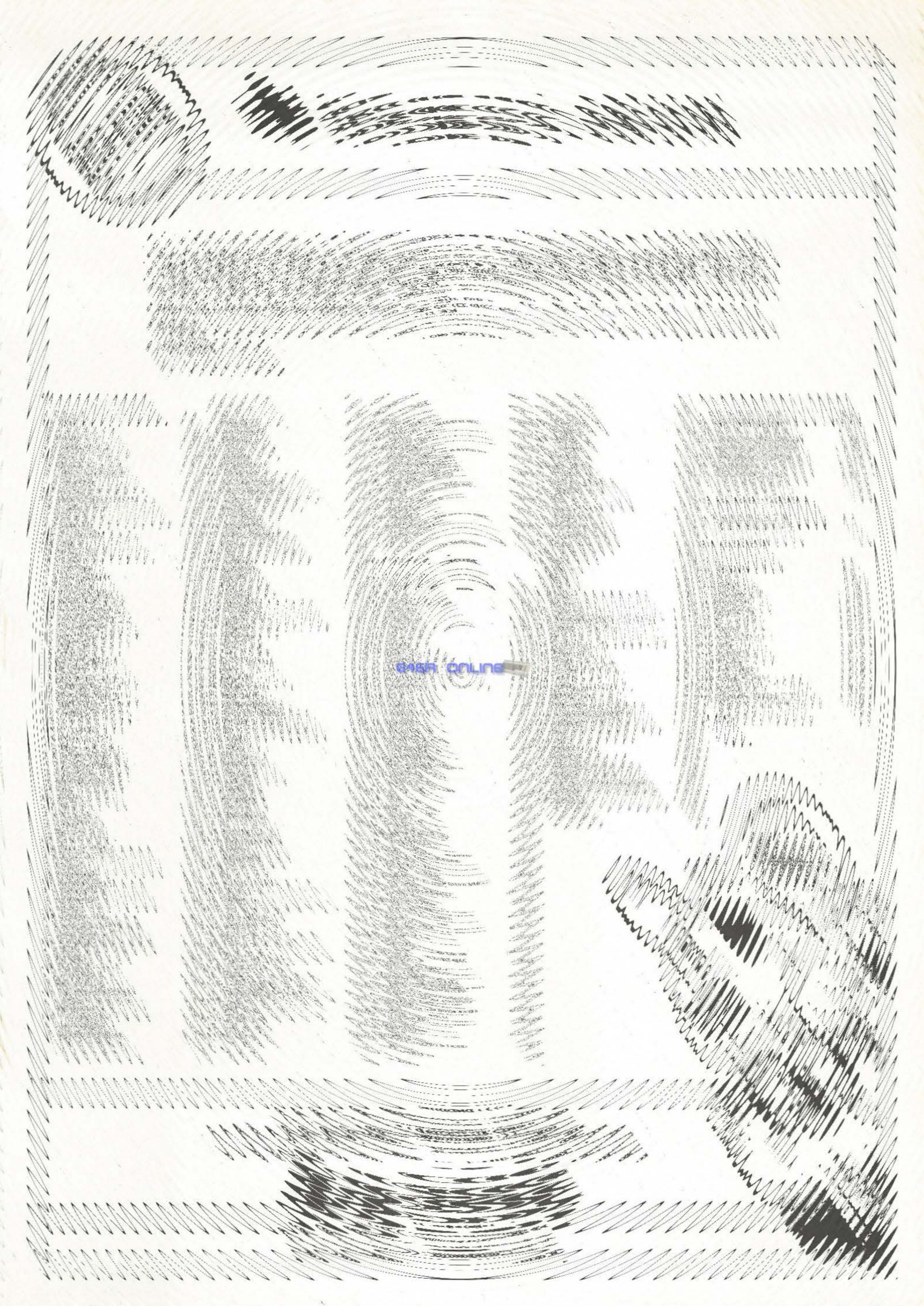
Spalten-nummer	0		1		2		3		4		5		6		7		8		9		10		11		Zahlen-codes
Werte	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	
Zeile 0																									
Zeile 1																									
Zeile 2																									
Zeile 3																									
Zeile 4																									
Zeile 5																									
Zeile 6																									
Zeile 7																									
Zeile 8																									
Zeile 9																									
Zeile 10																									
Zeile 11																									
Zeile 12																									
Zeile 13																									
Zeile 14																									
Zeile 15																									
Zeile 16																									
Zeile 17																									
Zeile 18																									
Zeile 19																									
Zeile 20																									

Hintergrundfarbe (transparent)	0 0	Mehrfarben-register 0	0 1	Sprite-Farbe	1 0	Mehrfarben-register	1 1

Setzen, Löschen und Abfragen einzelner Bits

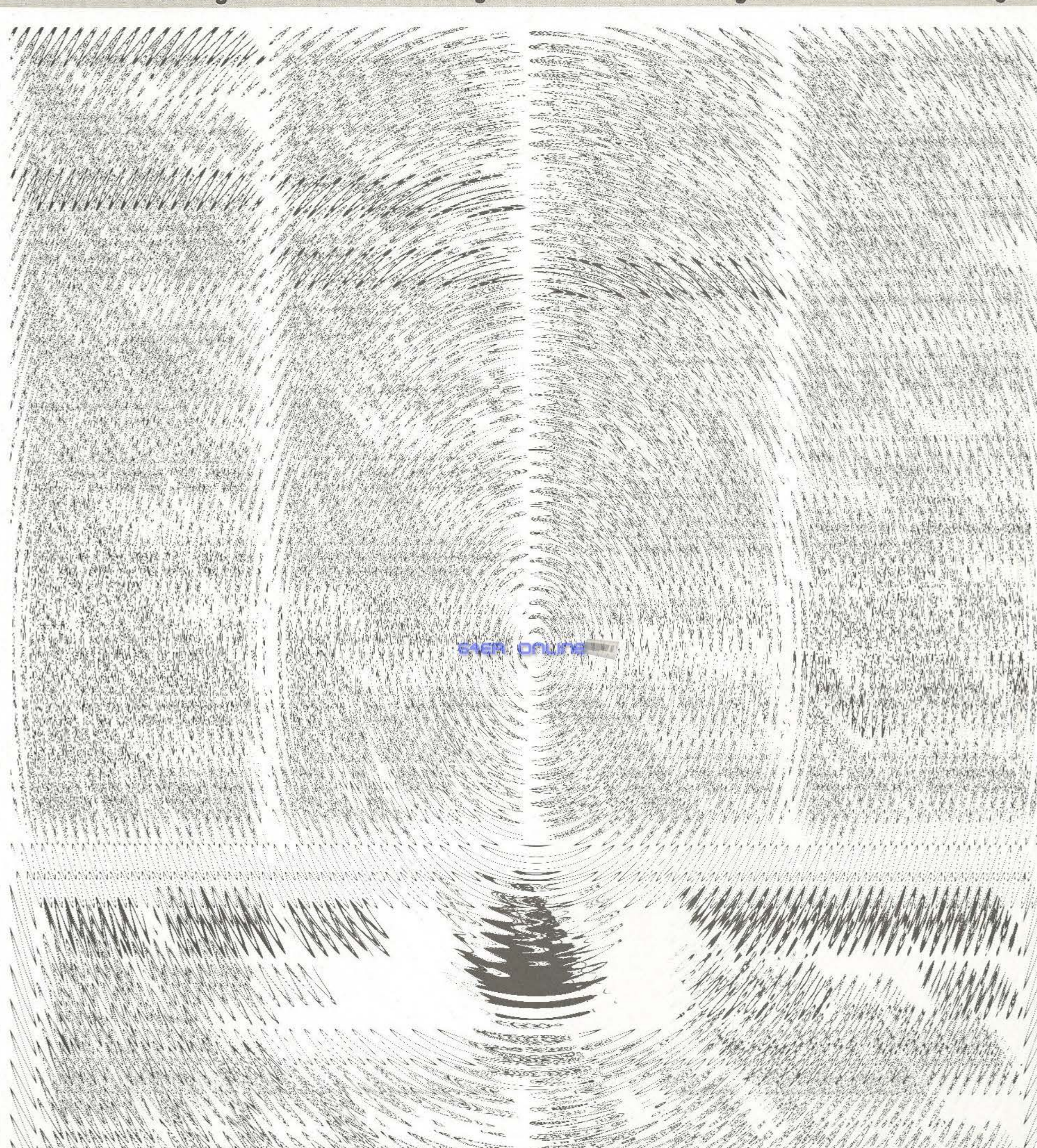
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Adresse hex	Adresse dezimal	Register
Sprite-Vergrößerung in Y-Richtung. 0 = normale Größe, 1 = doppelte Größe										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D017	53271	23
Startadresse Textbildschirm				Startadresse Zeichengenerator oder Hires-Bitmap			(unbe-nutzt)	$D018	53272	24
Interrupt-Flaggen-Register Interrupt				Lichtgrif-fel-Inter-rupt-Flagge	Sprite/ Sprite-Kollision	Sprite/ Hinter-grund-Kollision	Raster-Interrupt-Flagge	$D019	53273	25
Interrupt-Masken-Register Interrupt				Lichtgriffel-Interrupt-Maske	Sprite/ Sprite-Koll.-Maske	Sprite/-Hinter-grund-Kollision-Maske	Raster-Interrupt-Maske	$ D01A	53274	26
Sprite/Hintergrund-Prioritätenregister. 0 = Sprite hat Priorität, 1 = Hintergrund hat Priorität										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D01B	53275	27
Sprite-Mehrfarbmodus-Register. 0 = Normaldarstellung, 1 = Mehrfarbmodus-Darstellung										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D01C	53276	28
Sprite-Vergrößerung in X-Richtung. 0 = normale Größe, 1 = doppelte Größe										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D01D	53277	29
Sprite/Sprite-Kollision. 0 = keine Berührung, 1 = Berührung										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D01E	53278	30
Sprite/Hintergrund-Kollision. 0 = keine Berührung, 1 = Berührung										
Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0	$ D01F	53279	31
(unbenutzt)				Farbe des Bildschirmrahmens				$ D020	53280	32
(unbenutzt)				Hintergrundfarbe Nr.0 (normale Hintergrundfarbe)				$ D021	53281	33
(unbenutzt)				Hintergrundfarbe Nr. 1				$ D022	53282	34
(unbenutzt)				Hintergrundfarbe Nr. 2				$ D023	53283	35
(unbenutzt)				Hintergrundfarbe Nr. 3				$ D024	53284	36
(unbenutzt)				Sprite-Mehrfarben-Register Nr. 0				$ D025	53285	37
(unbenutzt)				Sprite-Mehrfarben-Register Nr. 1				$ D026	53286	38
(unbenutzt)				Sprite 0, Farbe				$ D027	53287	39
(unbenutzt)				Sprite 1, Farbe				$ D028	53288	40
(unbenutzt)				Sprite 2, Farbe				$ D029	53289	41
(unbenutzt)				Sprite 3, Farbe				$ D02A	53290	42
(unbenutzt)				Sprite 4, Farbe				$ D02B	53291	43
(unbenutzt)				Sprite 5, Farbe				$ D02C	53292	44
(unbenutzt)				Sprite 6, Farbe				$ D02D	53293	45
(unbenutzt)				Sprite 7, Farbe				$ D02E	53294	46

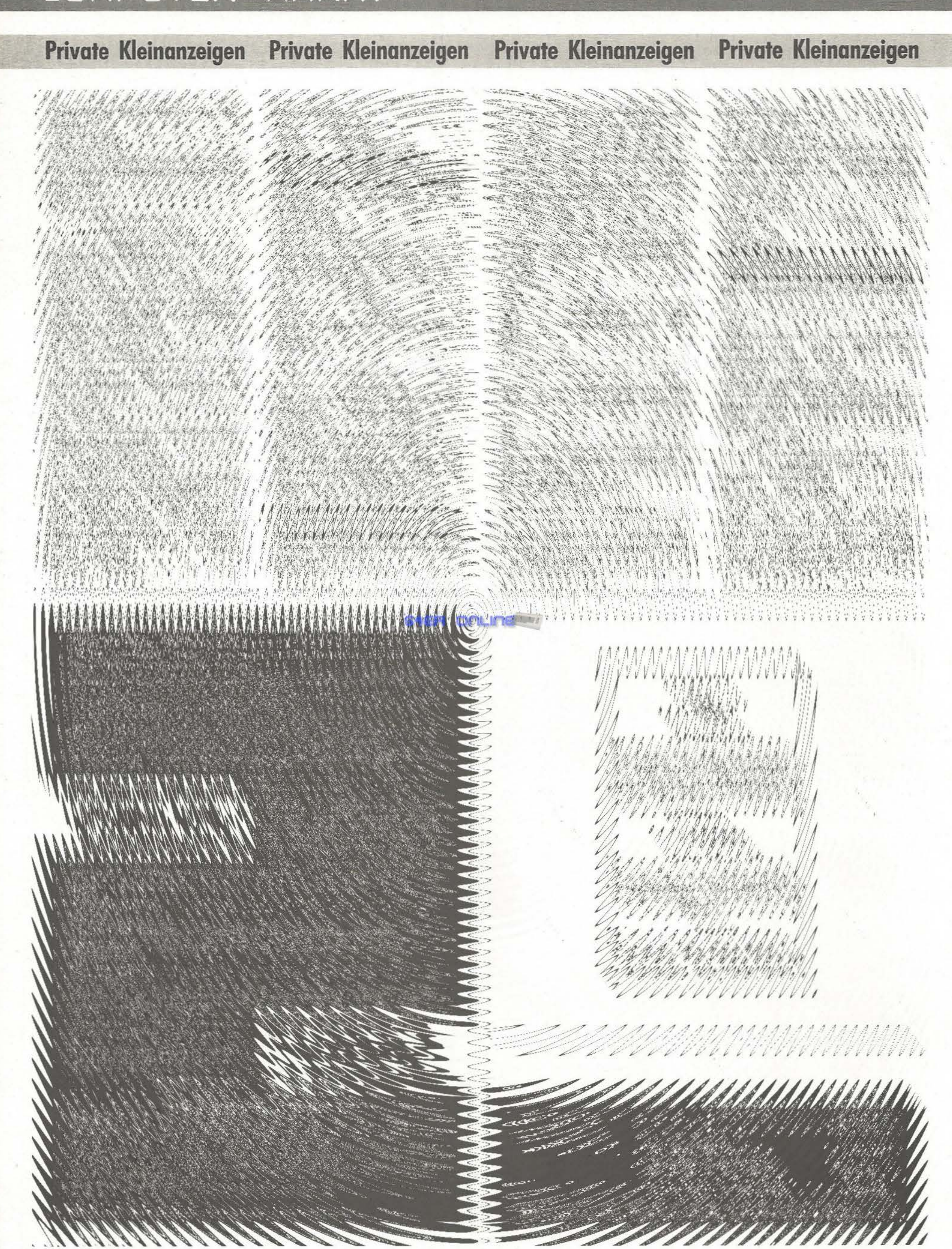


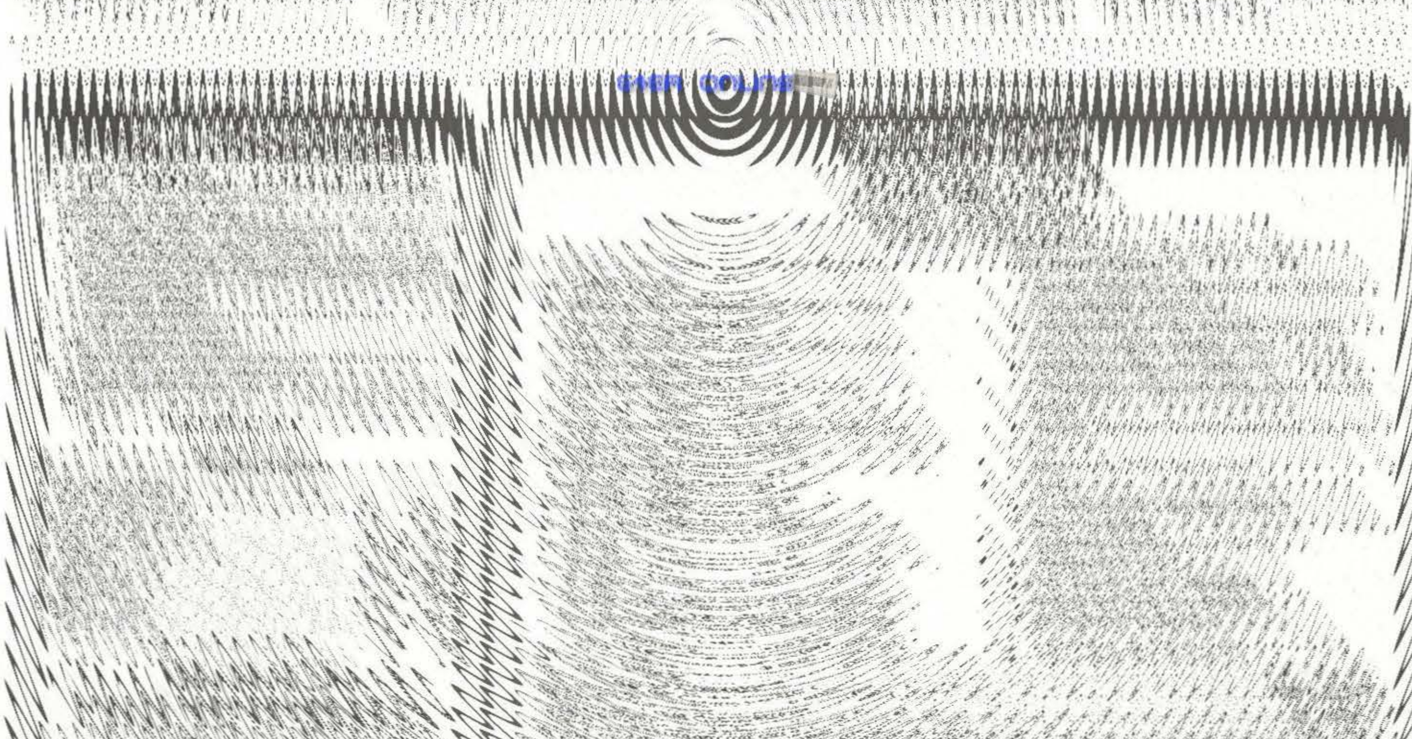


Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen



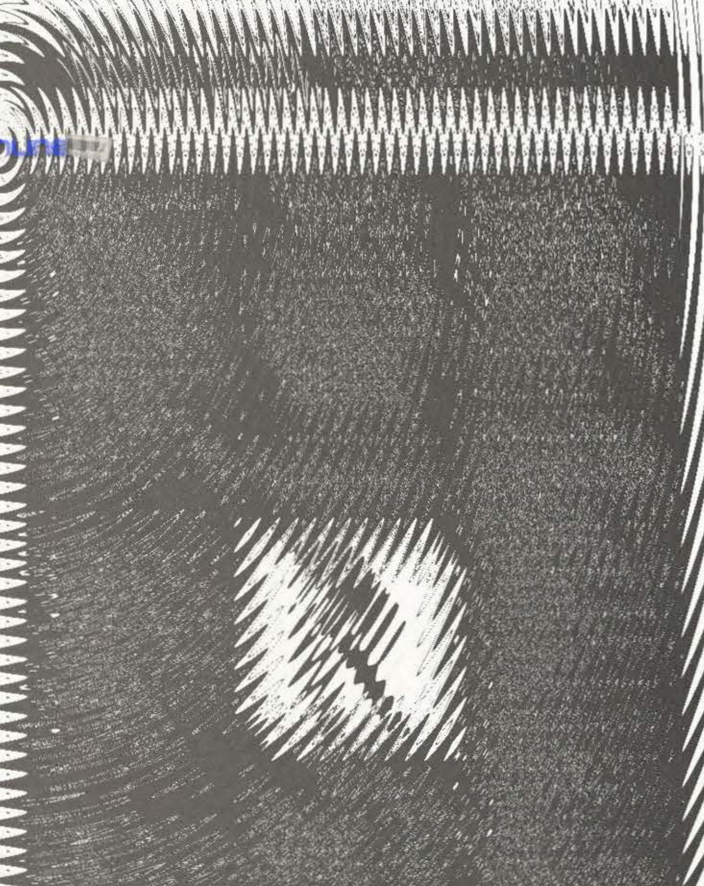






COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen

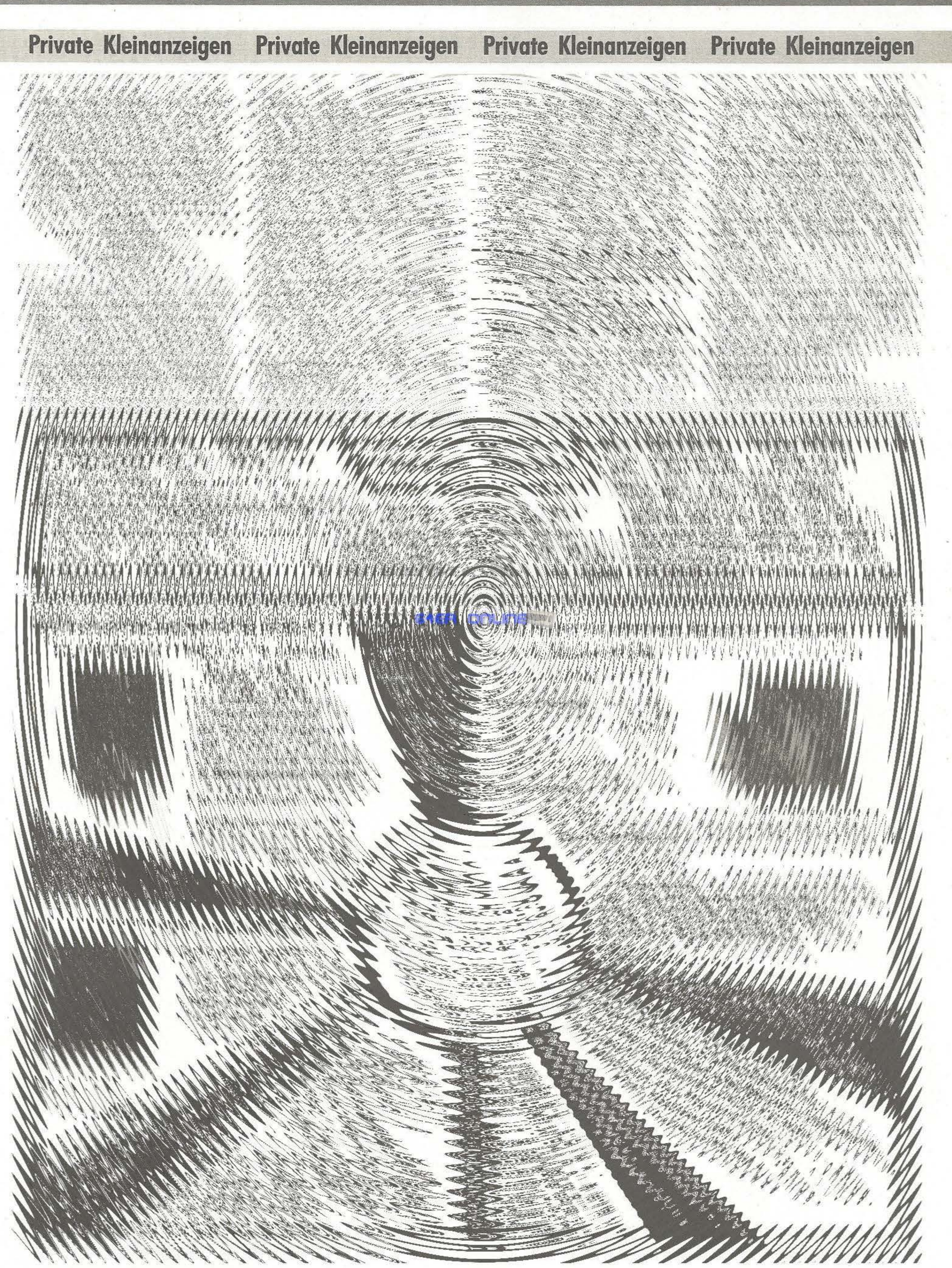


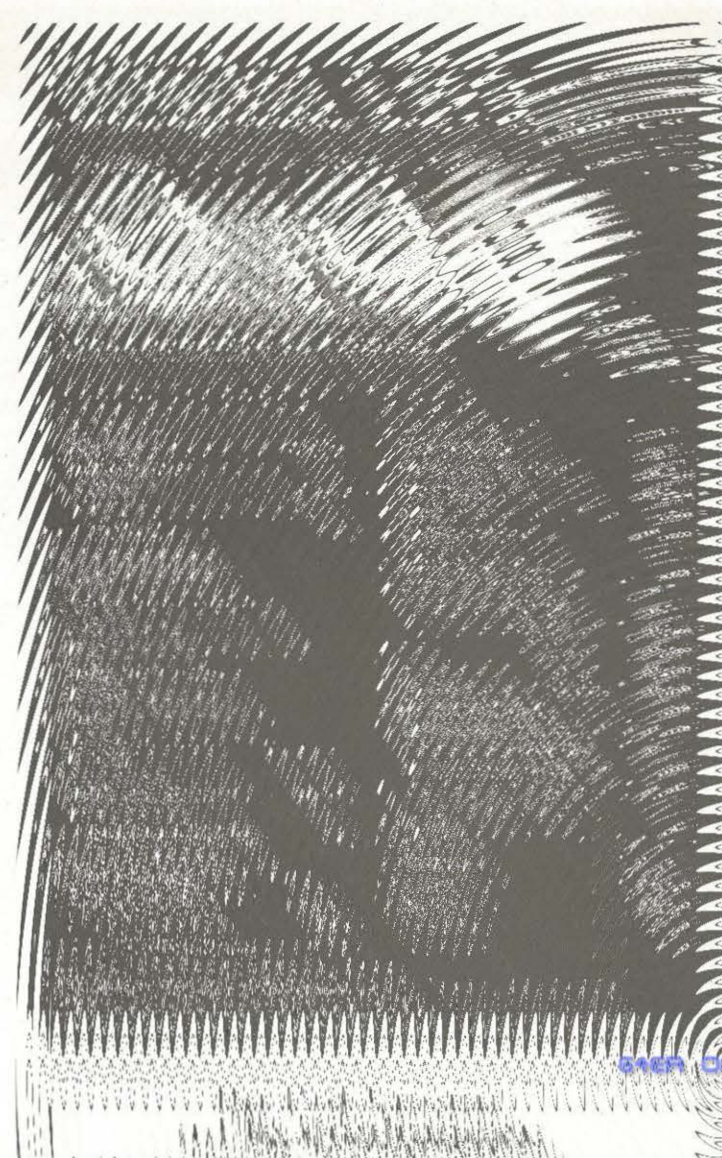


COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen

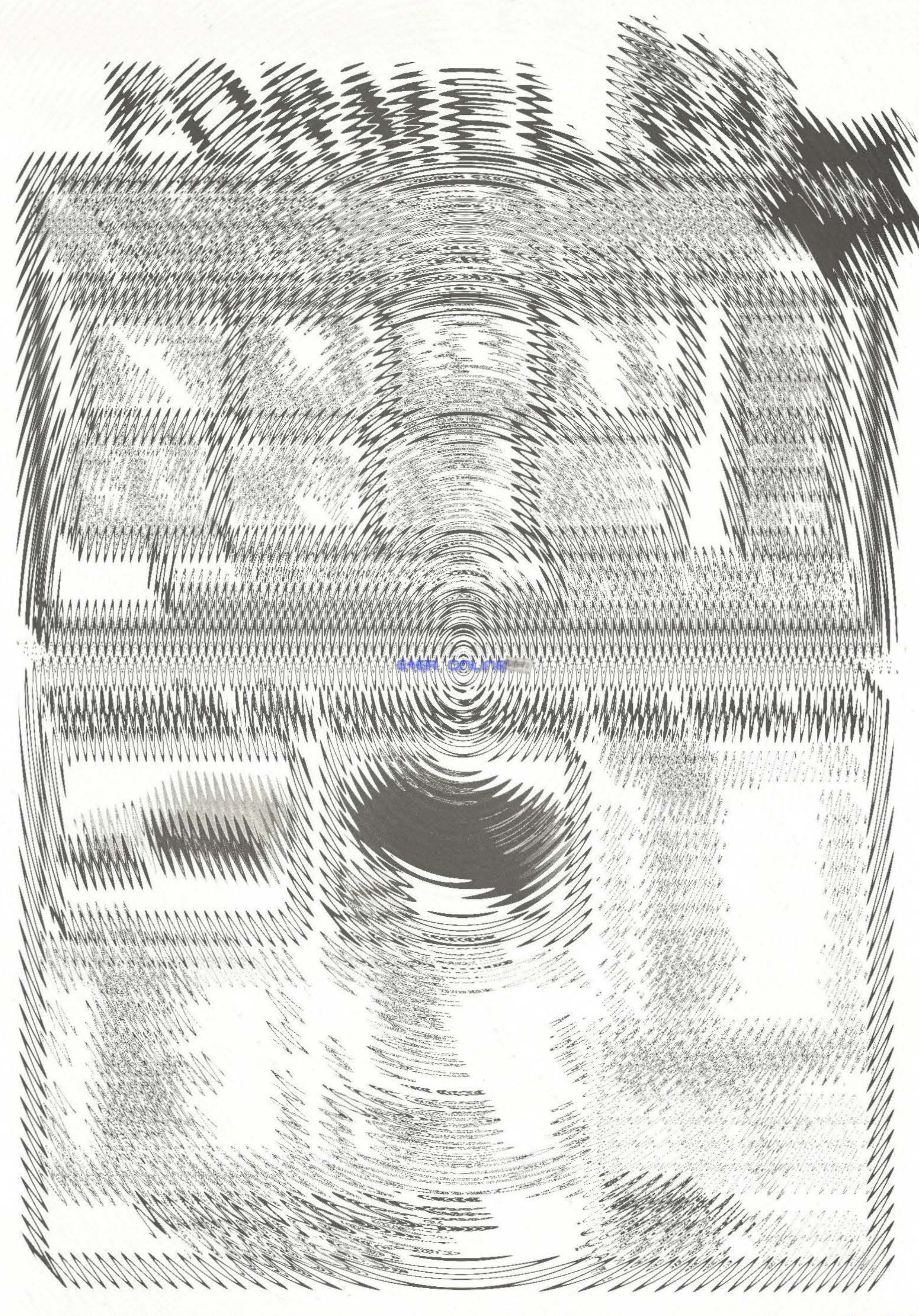
Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen

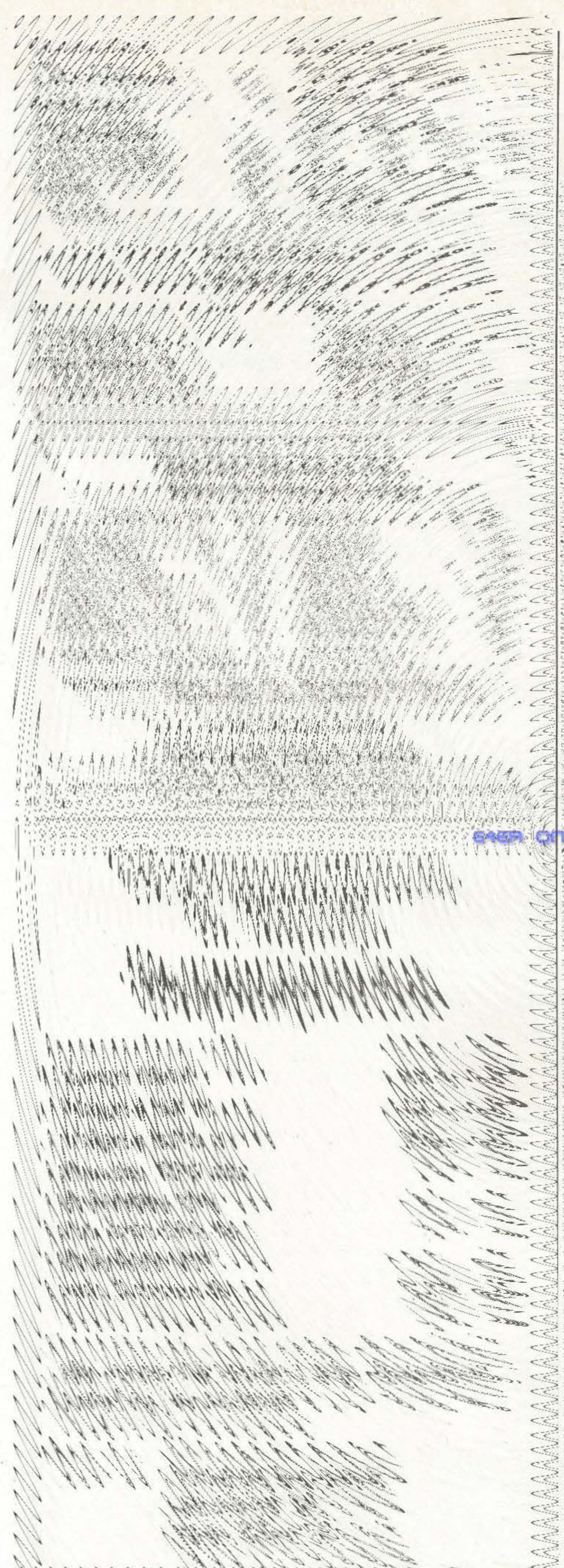




COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen



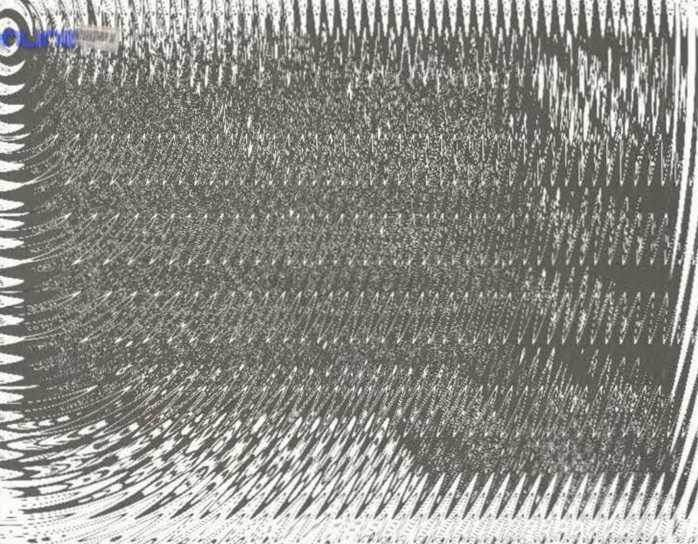


COMPUTER-MARKT

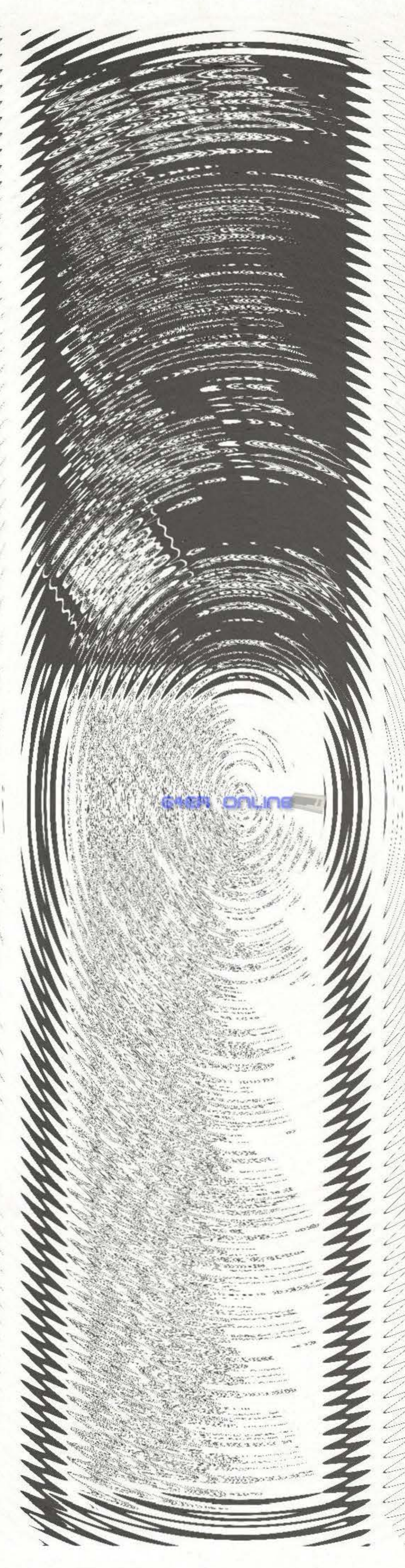
Private Kleinanzeigen Private Kleinanzeigen

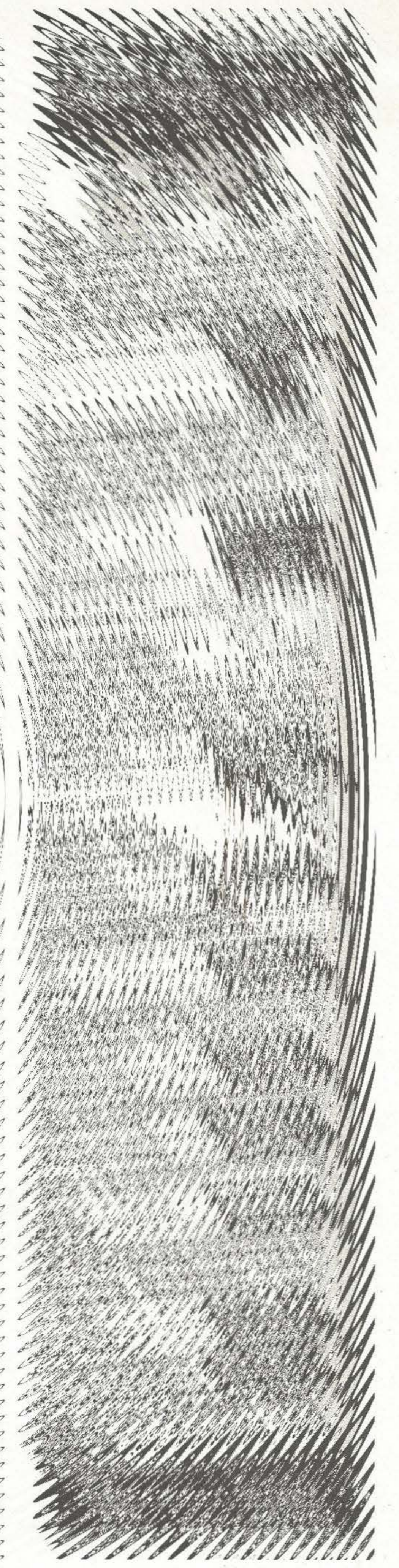
COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen



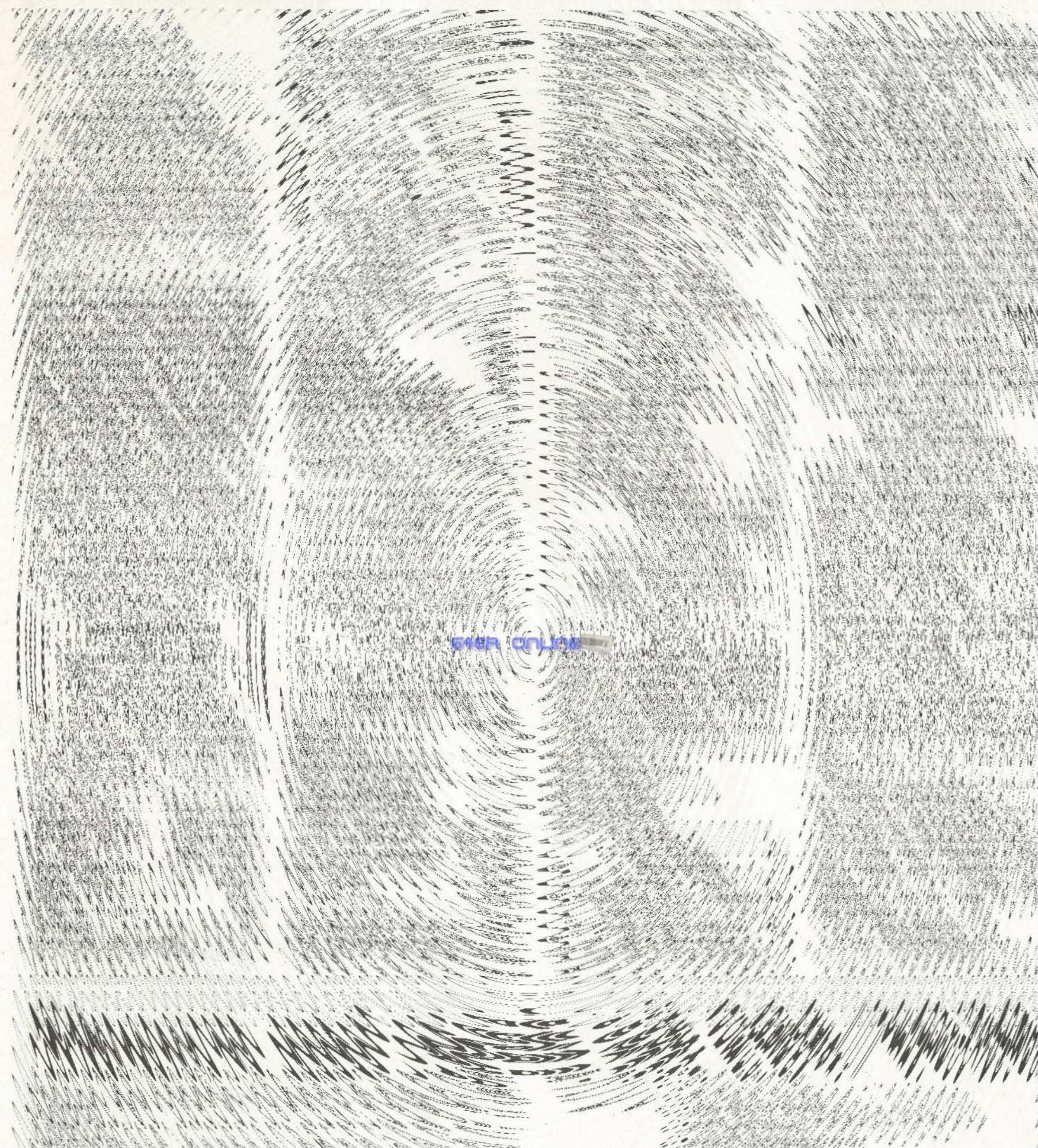
Gewerbliche Kleinanzeigen



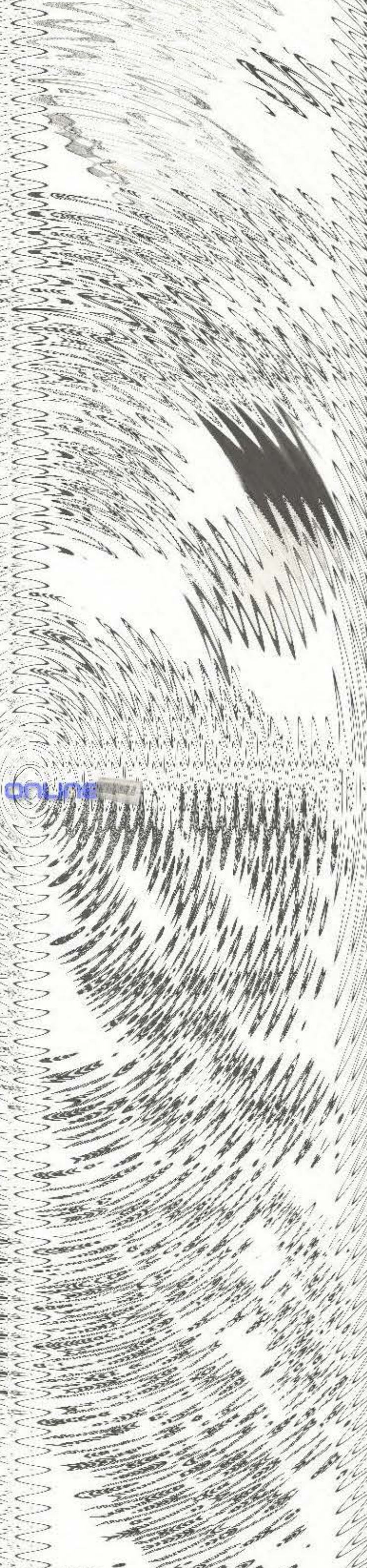


Gewerbliche Kleinanzeigen

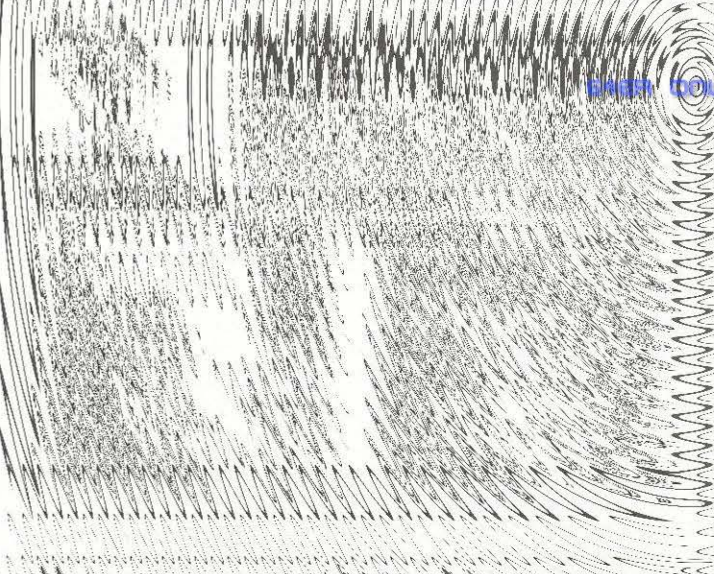
Gewerbliche Kleinanzeigen

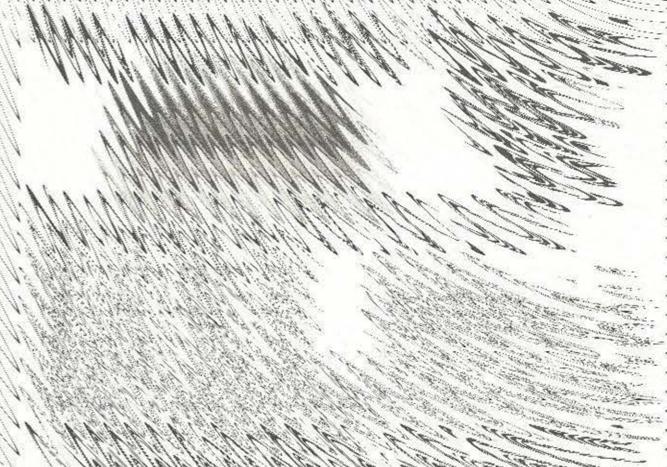


Gewerbliche Kleinanzeigen










1
2
3
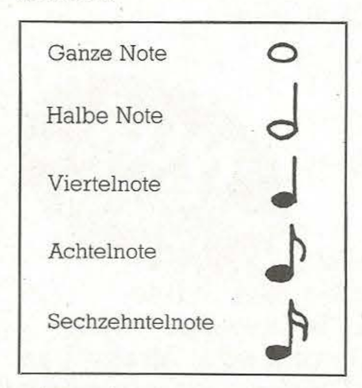
Dem Klang auf der Spur (Teil 9)

In diesem Teil wird gezeigt, wie man dreistimmige Musikstücke programmgesteuert, schnell und zeitexakt auf dem C 64 wiedergeben kann. So ganz nebenbei erfahren Sie eine Menge über die Interrupttechnik.

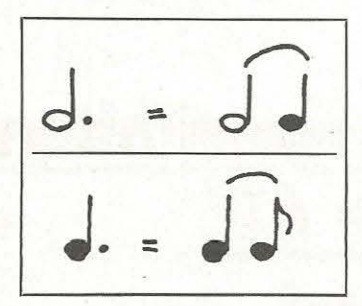
Zunächst einige Grundlagen zum Sequenzer. Unter einem Sequenzer versteht man ein Gerät oder Programm, das einen Synthesizer mit einer vorprogrammierten Tonfolge ansteuert. Die zentrale Rolle spielt dabei das genaue Einhalten eines programmierbaren Zeitmaßes.

Musikstücke werden üblicherweise in Takte von etwa 1 bis 4 Sekunden Länge eingeteilt. Am gebräuchlichsten ist der 4/4-Takt, der die Länge einer ganzen Note hat. Andere gebräuchliche Taktarten sind 2/4, 3/4, 6/8, 4/4, 5/4, 3/2. Diese Angaben betreffen allerdings nur die Zählweise der Takte und nicht das Tempo eines Musikstücks. So sind zum Beispiel 3/4- und 6/8-Takt bis auf die Zählweise vollkommen identisch.

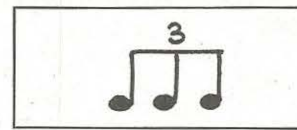
Die Notenlängen werden in Bruchteilen der ganzen Note angegeben:



Es kommen auch ungeradzahlige Vielfache dieser Notenlängen vor. Durch Punktierung kennzeichnet man die Verlängerung einer Note um die Hälfte ihrer ursprünglichen Länge:



Alle diese Notenlängen passen in ein Raster, welches eine ganze Note in 16 oder 32 gleiche Zeitabschnitte teilt. Es werden aber häufig auch sogenannte Triolen (Drittelnoten) eingesetzt. Zum Beispiel Achteltriolen,



das sind drei gleichlange Noten mit der Länge einer Viertelnote. Aus diesem Grund sollte das Zeitraster (die Anzahl der Zeitabschnitte, in die der Sequenzer eine ganze Note einteilt) auch den Faktor 3 enthalten. Ein sinnvolles Zeitraster ist zum Beispiel 96 (=3x32).

Das Tempo wird in der Musik in Schlägen pro Minute (beats per minute: bpm) gemessen. Ein Schlag entspricht dabei einer Viertelnote. Der sinnvolle Bereich für dieses Maß liegt bei etwa 40 bis 240 bpm. Beim schnellen Tempo 240 bpm dauert eine ganze Note genau eine Sekunde. Der Sequenzer muß dann 96 Schritte pro Sekunde ausführen.

Programmtechnik

Ein Sequenzer ist von der zu erbringenden Funktion her eigentlich ein sehr einfaches Programm. Seine Leistungen sind schnell aufgezählt:
— Tonhöhen steuern
— Triggerung der einzelnen Stimmen (GATE ON und GATE OFF)

Diese Steuerungen müssen zeitgenau und unabhängig voneinander für drei Stimmen erfolgen. Darüber hinaus wären einige Zusatzfunktionen sinnvoll:
— programmierbare Tempoänderungen
— programmierbare Soundwechsel
— programmierbare Änderung der Inhalte beliebiger Speicherplätze (Parameter-Änderung)

Es soll zunächst ein einfacher Basis-Sequenzer entwickelt werden, der sich dann leicht um die genannten Zusatzfunktionen erweitern läßt. Die Erweiterungen sollen über Vektoren, also ohne Änderung des Grundprogramms, an dieses angeschlossen werden können.

Ein Sequenzer ist, ähnlich wie der in dieser Reihe veröffentlichte Modulator, ein Programm, das in regelmäßigen Zeitabständen eine Leistung erbringen muß. Der Aufruf per Interrupt, ausgelöst durch einen Zeitgeber, bietet sich also auch hier an. Jeder CIA (Complex Interface Adapter) ist mit zwei 16-Bit-Timern ausgestattet, die sich für diese Aufgabe eignen. Timer A in CIA1 wird bereits für den Systeminterrupt eingesetzt. Ein Systeminterrupt findet konstant 60-mal pro Sekunde statt und kann mit dem Aufruf eines Modulatorschritts gekoppelt werden.

Musik per Interrupt

Die Aufruffrequenz der Sequenzer-Schritte soll dagegen im Bereich von zirka 20-100 Hz, abhängig vom Tempo des Musikstücks, variabel sein. (Man erinnere sich: 240 bpm entsprechen 96 Hz bei einem Zeitraster von 96 Schritten pro ganzer Note.) Das legt den Einsatz eines weiteren unabhängigen Timers nahe. In Frage kommen dafür:
Timer B in CIA1 (IRQ)
Timer A in CIA2 (NMI)
Timer B in CIA2 (NMI)

Die Auswahl des Timers ist willkürlich. Im vorliegenden Programm wird Timer B in CIA1 eingesetzt. Dadurch bleiben die Timer in CIA 2 noch vollkommen frei für Zwecke, die nichts mit der Musikprogrammierung zu tun haben müssen. Da nun Timer A und Timer B beide unabhängig voneinander Interrupts auslösen können, muß die angesprungene Interrupt-Serviceroutine die Interrupt-Quelle ermitteln, also feststellen, welcher Timer den Interrupt ausgelöst hat und abhängig davon weiterverzweigen. Zu diesem Zweck wird im sogenannten Interrupt-Control-Register (ICR) $DC0D bei einem Timer-A-Interrupt Bit 0 und bei einem Timer-B-Interrupt Bit 1 gesetzt.

Programmierung des CIA

Zur Steuerung von CIA-Interrupts dient das schon erwähnte Interrupt-Control-Register (ICR) $DC0D. Dieses Register hat zwei Funktionen, je nachdem, ob schreibend oder lesend darauf zugegriffen wird. Bei Lesezugriff zeigt es an, ob, und wenn ja, woher ein Interrupt ausgelöst wurde. Zugleich wird das Register gelöscht und die Interrupt-Anforderung zurückgenommen (Die IRQ-Leitung geht von low auf high). Die Bits 0-4 sind dabei verschiedenen Interruptquellen zugeordnet. Uns interessieren hier nur die Bits 0 und 1, welche zu den Timer-Interrupts gehören. Durch einen Schreibzugriff wird dagegen ein Masken-Register angesprochen. Damit lassen sich die Interruptquellen einzeln freigeben oder sperren. Die Bits 0-4 kann man einzeln setzen oder zurücksetzen. Ist im geschriebenen Byte Bit 7 gesetzt, wird jedes mit einer 1 beschriebene Bit gesetzt, während die anderen Bits unverändert bleiben. Ist Bit 7 rückgesetzt, so wird jedes mit einer 1 beschriebene Bit zurückgesetzt, während die anderen Bits wieder unverändert bleiben. Gesetzte Bits ermöglichen eine Interrupterzeugung durch die jeweilige Quelle. Die Freigabe der Interrupterzeugung durch Timer B sieht also so aus:

```
LDA #%10000010
STA $DD0D        ;ICR Bit 1 setzen
```

Der Timer selbst wird durch drei Register gesteuert.

Das Registerpaar TIMER B ($DC06/$DC07) liefert bei Lesezugriff den aktuellen 16-Bit-Zählerstand. Dieser Wert wird kontinuierlich heruntergezählt. Bei Erreichen von Null stoppt der Timer entweder (One-Shot-Mode) oder lädt einen Wert aus einem Timer-Latch (Latch = Zwischenspeicher) nach und zählt von neuem herunter (Continous Mode). Bei diesem Timer-Unterlauf wird ein Interrupt erzeugt, wenn Bit 1 im ICR gesetzt ist. Ein Schreibzugriff auf TIMERA bezieht sich dagegen auf das 16-Bit-Latch. Mit dem Latch-Wert kann man die Zeit zwischen zwei Interrupts im Bereich von 1 bis 65535 Mikrosekunden steuern.

Das Register CRB (Control Register B, $DC0F) steuert die Betriebsart des Timers (Start/Stop, One Shot/Continous, u.a.)
Durch

```
LDA #%00010001
STA $DD0E
```

wird der Zählerstand mit dem Latch-Wert geladen und der Timer gestartet.

Die Interrupt-Service-Routine

Sie fragt zunächst ab, ob der Interrupt von Timer A (Systeminterrupt, Modulatorschritt) oder von Timer B (Sequenzerschritt) kommt. Bei einer möglichen gleichzeitigen Interruptanforderung durch beide Timer, wird der Timer-B-Interrupt bevorzugt behandelt. Das hat folgende Gründe:

— Für ein exaktes Sequenzer-Timing sollten anzuspielende Noten möglichst wenig verzögert werden.

— Die Abarbeitung eines Sequenzer-Schritts benötigt viel weniger Rechenzeit als ein Modulatorschritt (zeitaufwendige Multiplikationen) oder eine Systeminterrupt-Behandlung.

— Die Aufruffrequenz kann bei den Sequenzer-Schritten sehr hoch sein (96 Hz bei 240 bpm, aber auch über 200 Hz sind technisch leicht möglich).

Da das ICR beim Lesen gelöscht wird, muß sein Inhalt zwischengespeichert werden, damit beim Auftreten von zwei Interrupts die Behandlung des niedriger priorisierten Timer-A-Interrupts nachgeholt werden kann.

Bei Auftreten eines Interrupts wird immer das Interrupt-Bit im CPU-Statusregister gesetzt, damit die CPU nicht gleich wieder unterbrochen werden kann. Da die IRQ-Leitung so lange auf Low-Pegel bleibt, bis die CPU durch Auslesen des CIA-ICR die Interruptanforderung löscht, würde sich das System ohne gesetztes Interrupt-Bit durch einen Dauerinterrupt aufhängen. Es steht dem Programmierer allerdings frei, nach dem Auslesen des ICR das Interrupt-Bit durch den Befehl CLI (Clear Interrupt-Flag) zurückzusetzen, um damit das Programm wieder unterbrechbar zu machen. Beim vorliegenden Programm bleibt bei einem Sequenzer-Schritt das Interrupt-Bit gesetzt, während es zur Abarbeitung eines Timer-A-Interrupts rückgesetzt wird. Dadurch kann die CPU auch dann durch einen Timer-B-Interrupt unterbrochen werden.

Das Betriebssystem und das Programm Modulator machen beide intensiven Gebrauch von der Zero-Page. Die Inhalte der Zero-Page-Speicherplätze dürfen von einem interruptgetriebenen Programm nicht verändert werden. Das Sequenzer-Programm belegt daher nur zwei Zero-Page-Speicherplätze ($FE,$FF). Ihre Inhalte werden bei Programmbeginn zwischengespeichert und bei Programmende restauriert.

Die verwendeten Datenstrukturen

Um ein Musikstück in eine computergerechte Form zu bringen, muß man im wesentlichen die Tonhöhe und die Länge der einzelnen Noten codieren. Beim Einsatz mehrerer, verschieden klingender Stimmen, muß man außerdem jede Note eindeutig einer Stimme zuordnen. Die hier verwendete Datenstruktur (Bild 1) verfolgt mit ihrem etwas komplizierten Aufbau zwei Ziele:
— Sparsamer Umgang mit dem Speicher
— Gute Editiermöglichkeiten. (Ein Editorprogramm in Basic folgt in der nächsten Ausgabe)

Tracks

Die Steueranweisungen werden für die drei Stimmen getrennt in drei sogenannten Tracks (Tonspuren) gespeichert. Ein Track ist eine zusammenhängende Folge von 1-Byte-Kommandos. Das häufigste Kommando dürfte das Ton-Kommando sein. Die Tonhöhe wird aus einer Oktav-Nummer und einer Tonnummer (siehe Bild 2), die in den beiden Nibbles (= Halbbytes) eines Bytes stehen, ermittelt. Das Programm benötigt dazu lediglich eine Tabelle der Frequenzen der höchsten Oktave. Die Frequenzen der niedrigeren Oktaven werden durch Teilung durch Zweierpotenzen errechnet. Eine Division durch 2 wird durch einen einfachen Rechts-Shift realisiert. Die Dauer des Tones ist nicht Bestandteil des Ton-Kommandos. Sie wird durch das Zeit-Kommando voreingestellt. Da häufig mehrere Töne mit gleicher Länge aufeinanderfolgen, genügt ein einziges Zeit-Kommando (ein oder zwei Bytes), um die Tonlänge (siehe Bild 3) einzustellen. Dabei wird zwischen einer GATE-ON- und einer GATE-OFF-Phase unterschieden, deren Längen zusammengenommen die gewünschte Tonlänge ergeben.
Beispiel: GATE-ON-Zeit = 5
GATE-OFF-Zeit = 7
Gesamtzeit = 12

Das entspricht einer kurz angeschlagenen Achtelnote (bei 96 Zeitschritten pro ganzer Note). Die GATE-ON-Zeit ist im Bereich 1-96, die GATE-OFF-Zeit im Bereich 0-30 einstellbar. Der Sequenzer setzt nach Ablauf der GATE-ON-Zeit das GATE-Bit der entsprechenden Stimme im SID zurück und wartet dann die GATE-OFF-Zeit ab. Ist diese 0, so wird natürlich sofort der nächste Ton gespielt. Man kann aber auch explizit Pausen programmieren (Code $EF). Ihre Länge ist die Summe aus GATE-ON- und GATE-OFF-Zeit.

Der Code $00 ist zur Kennzeichnung für das Track-Ende vorgesehen. Die Codes $F8 bis $FF sind für Sonderfunktionen reserviert, die für eine spätere Erweiterung des Sequenzers gedacht sind. Angesprungen werden sie über eine Tabelle von Vektoren, die im Moment

Tonnummer	Note
0	c
1	cis = des
2	d
3	dis = es
4	e
5	f
6	fis = ges
7	g
8	gis = as
9	a
10	ais = b
11	h

Bild 2. Tonnummer und Note

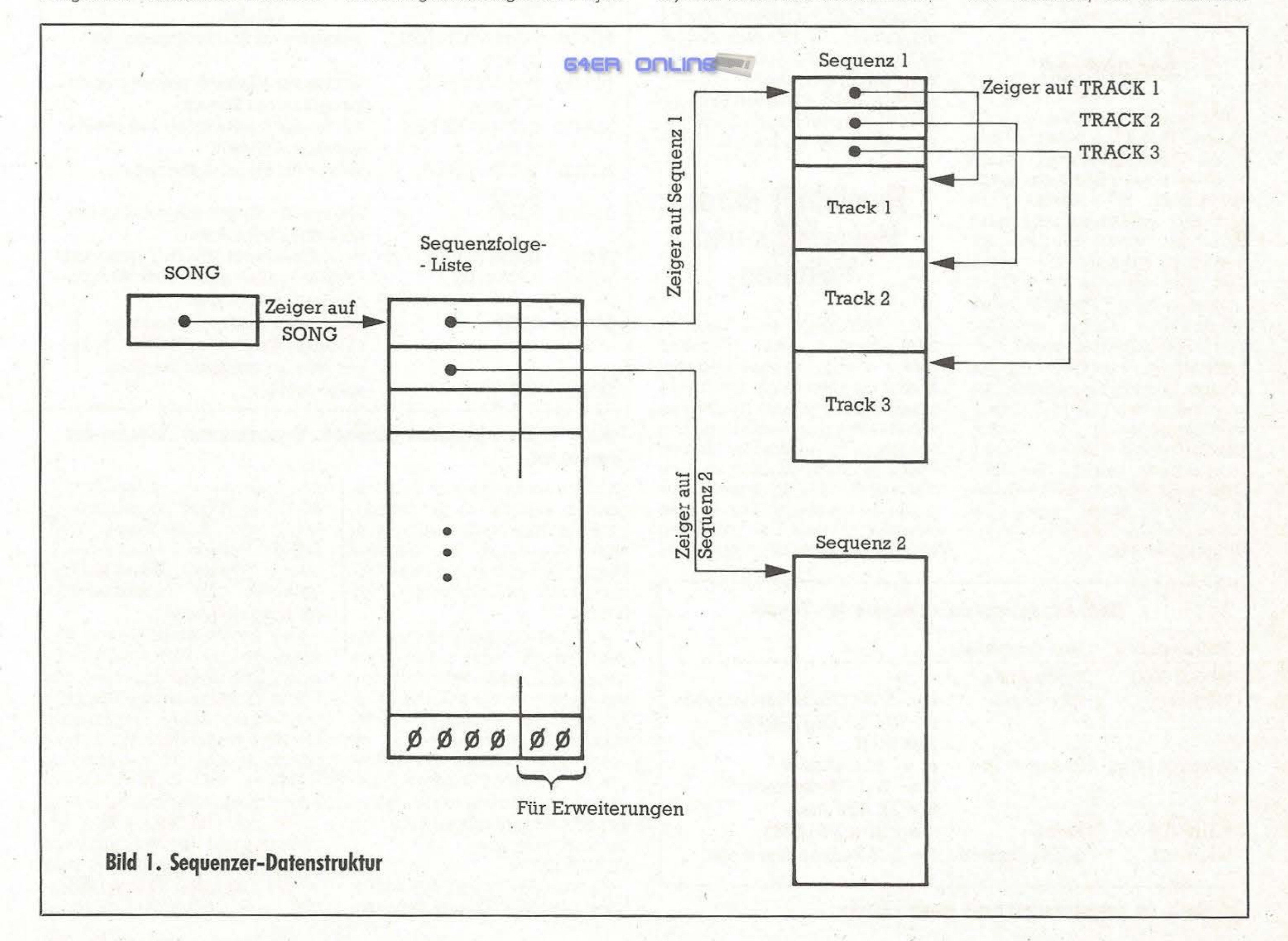


Bild 1. Sequenzer-Datenstruktur

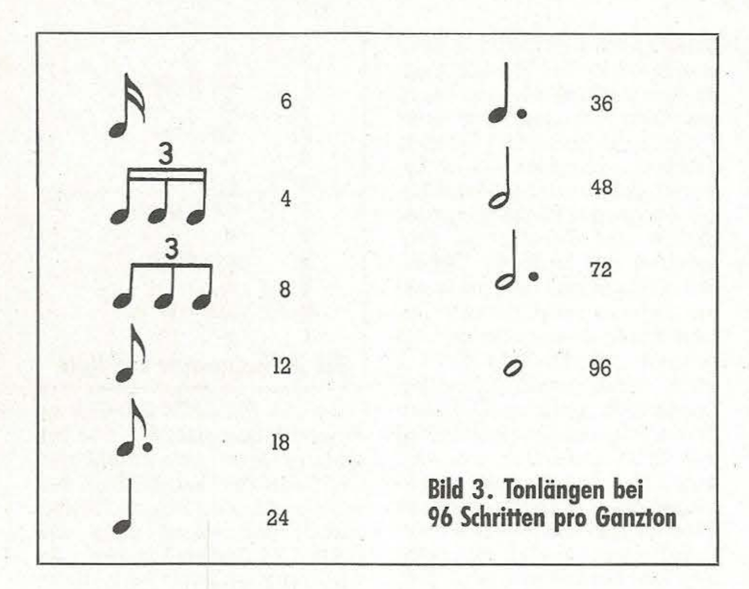


Bild 3. Tonlängen bei 96 Schritten pro Ganzton

nur in den Programmteil zur Ausführung des nächsten Kommandos führen, also nichts bewirken. Sinnvolle Sonderfunktionen sind:
— Änderung von Soundparametern
— Wahl eines ganzen Parametersatzes (Soundwechsel) im Zusammenhang mit dem Programm Modulator
— Tempowechsel

Diese Sonderfunktionen werden den Sequenzer in der nächsten Folge ergänzen.

Sequenzen

Für jede der drei Stimmen gibt es eine Folge von Kommandos, einen Track. Die drei Tracks werden zu einer Sequenz zusammengefaßt. Eine Sequenz ist hier ein zusammenhängender Abschnitt eines Musikstücks, der einen einzigen Ton, einen Takt oder auch das ganze Stück umfassen kann. Den drei Tracks gehen drei Zeiger auf die Track-Startadressen voran. Obwohl es sich aus Gründen der Übersichtlichkeit empfiehlt, die Sequenzen wie in Bild 1 zusammenhängend in der Folge Zeiger-Track 1, -Track 2, -Track 3 zu speichern, besteht dazu kein Zwang. Es müssen lediglich die drei Track-Zeiger einer Sequenz und die Tracks in sich zusammenhängen.

Sequenzfolge-Liste

Um eine Sequenz zu wiederholen, muß man sie nicht zweimal programmieren, sondern kann sie wie ein Unterprogramm mehrmals aufrufen. Die Sequenzfolge-Liste enthält dazu die Startadressen der Sequenzen in der Reihenfolge, in der diese gespielt werden sollen. Dabei können die gleichen Adressen natürlich mehrfach auftreten. Unter der Startadresse einer Sequenz wird hier die Adresse des Zeigers auf Track 1 verstanden. Die Sequenzfolge-Liste enthält für jede Sequenz außer dem Zeiger noch ein drittes Byte, das für spätere Erweiterungen vorgesehen ist. Drei Nullen schließen die Liste ab.

Flexibilität durch Steuerflags und Vektoren

Im Normalfall wird man die drei Tracks einer Sequenz gleich lang programmieren. Macht man dagegen die Tracks unterschiedlich lang, so wiederholt das Programm die kürzeren Tracks so lange, bis der längste Track zu Ende gespielt ist. Erst dann geht das Programm zur nächsten Sequenz über. Dieses Verhalten kann bei manchen Musikstücken nützlich sein. Das MSE-Listing 2 enthält einen Musik-Datensatz, bei dem in der zweiten Sequenz der dritte Track aus nur vier Tönen besteht, die fortlaufend wiederholt werden.

Normalerweise hält der Sequenzer an, wenn alle Sequenzen gemäß Sequenzfolge-Liste durchgespielt sind. Nach dem Anhalten wird auch der Interruptvektor auf seinen ursprünglichen Wert zurückgestellt. Eine 1 im Flag REPMODUS bewirkt, daß das ganze Stück endlos wiederholt wird.

Eine 1 im Flag SEQMODUS bewirkt, daß die aktuelle Sequenz endlos wiederholt wird. Auch hier ist der längste Track der Sequenz maßgeblich.

Eine 1 im Flag LEGATO bewirkt, daß die GATE-Bits in den SID-Steuerregister nicht zurückgesetzt werden. Dadurch klingen die Töne gebunden. Dazu muß allerdings ein Sustain-Pegel ungleich Null eingestellt sein, sonst ist überhaupt nichts hörbar.

An allen wichtigen Stellen des Sequenzers wird der Programmfluß über Vektoren weitergeleitet. Damit soll die Möglichkeit, das Programm nachträglich leicht zu erweitern, offengehalten werden. Die Vektoren für die acht Sonderfunktionen wurden schon erwähnt. Außer diesen acht gibt es noch drei weitere Vektoren:

TONVEKTOR

Er führt das Programm weiter, nachdem die Frequenz für einen Ton-an-Befehl ermittelt wurde. Im vorliegenden Programm wird der Frequenzwert direkt in den SID geschrieben. Bei einem Einsatz zusammen mit dem Modulator muß die Frequenz dagegen in ein Modulator-Register geschrieben werden.

EXTRAVEKTOR

Über diesen Vektor kann man weitere Aktionen an einen Sequenzer-Schritt anhängen. Denkbar wäre zum Beispiel die Anzeige der gespielten Noten auf dem Bildschirm in Realtime.

IRQAVEKTOR

Führt zum Systeminterrupt $EA31. Dieser Vektor muß beim Einsatz mit dem Modulator auf die Startadresse des Modulatorschrittes zeigen.

Das vorliegende Sequenzerprogramm (Listing 1) belegt den Speicherbereich $C480-$C778. $C480 = 50304 ist gleichzeitig auch die Startadresse (SYS 50304). Tabelle 1 faßt die wichtigsten Routinen, Variablen und Vektoren des Sequenzer-Programms zusammen.

Routinen, Variablen, Vektoren

Adresse	Name	Beschreibung
$C480	JMP TEST	Teststart, Zeiger initialisieren, die wichtigsten SID-Parameter setzen, Sequenzer starten.
$C49E	SIDCR (3 Byte)	SID-Control-Register-Bytes mit zurückgesetztem GATE-Bit
$C4A8	LEGATO (1 Byte)	Flag 0 = normaler Betrieb 1 = kein GATE-OFF
$C4A9	SEQMODUS (1 Byte)	Flag 0 = ganzes Stück spielen 1 = Sequenz wiederholen
$C4AA	REPMODUS (1 Byte)	Flag 0 = Stück einmal spielen 1 = Stück immer wieder spielen
$C4C3	FUNCTION (8*2 Byte)	Vektoren für Sonderfunktionen
$C4D3	TONVEKTOR (2 Byte)	Vektor zur Weiterverarbeitung der Frequenz bei Ton an
$C4D5	EXTRAVEKTOR (2 Byte)	Vektor für Zusatzaktion bei jedem Sequenzer-Schritt
$C4D7	IRQAVEKTOR (2 Byte)	Vektor für Timer-A-Interrupt
$C4D9	START	Sequenzer starten (Es werden keine Zeiger initialisiert)
$C51E	IRQSERVICE	Anlaufpunkt für alle IRQ-Interrupts
$C67D	NEXTAKT	Dorthin sollten alle Sonderfunktionen zurückspringen
$C716	STOP	Sequenzer unterbrechen/ausschalten. Er kann mit START jederzeit wieder gestartet werden.
$C739	TEST	siehe $C480

Tabelle 1. Die wichtigsten Routinen, Varianten und Vektoren des Sequenzers.

Das Programmieren von Musikstücken mit Hilfe der Tabelle 2 ist noch etwas mühsam. Ein Editor in der nächsten Ausgabe wird diese Arbeit erleichtern. Mit dem Datensatz aus Listing 2 (»Kobold« aus den »Lyrischen Stücken« von Edvard Grieg) kann man den Sequenzer testen.

Die Verschmelzung des Sequenzers und des Modulators zu einer funktionellen Einheit wird in der nächsten Folge behandelt. (Thomas Krätzig/tr)

Kommandoformate innerhalb der Tracks

Kommando	Interpretation	
%0000 0000	Track-Ende	
%0ttt tttt	Zeitvorgabe	t = 1...96 GATE-ON-Zeit:=t-m t = 97...127 GATE-OFF-Zeit:=t-97
%1mmm nnnn	nächster Ton	m = 0...6 Oktave n = 0...11 Tonnummer n = 12...15 Pause
%1110 1111	Pause	(Standardcode $EF)
%1111 1fff	Sonderfunktion	f = 0...7 Funktionsnummer

Tabelle 2. So programmiert man einen »Track«.

```
programm : sequencer           c480 c779
-------------------------------------

c480 : 4c 39 c7 ff 00 00 00 c8  ec
c488 : 0c c8 e0 c8 fd c8 20 c9  84
c490 : 43 c9 18 18 18 00 00 00  43
c498 : 0d 0d 0d 00 00 00 20 20  30
c4a0 : 20 01 02 04 00 00 00 00  42
c4a8 : 00 00 01 1e 86 18 8e 8b  27
c4b0 : 96 7e 9f fa a8 06 b3 ac  af
c4b8 : bd f3 c8 e6 d4 8f e1 f8  c1
c4c0 : ee 2e fd 7d c6 7d c6 7d  63
c4c8 : c6 7d c6 7d c6 7d c6 7d  1c
c4d0 : c6 7d c6 fc c6 95 c5 31  38
c4d8 : ea ad 14 03 8d d7 c4 ad  04
c4e0 : 15 03 8d d8 c4 a9 1e 8d  22
c4e8 : 14 03 a9 c5 8d 15 03 a9  82
c4f0 : e0 8d 06 dc a9 2e 8d 07  04
c4f8 : dc a9 82 8d 0d dc a9 11  7c
c500 : 8d 0f dc 60 ae 84 c4 bd  f6
c508 : 8d c4 85 ff bd 8c c4 85  b7
c510 : fe a0 00 b1 fe fe 8c c4  38
c518 : d0 03 fe 8d c4 60 ad 0d  fb
c520 : dc 48 29 02 d0 03 4c a1  45
c528 : c5 a5 fe 48 a5 ff 48 a5  4f
c530 : 01 29 fe 85 01 ad 0d dc  a2
c538 : a2 0e 8e 85 c4 a2 04 8e  c4
c540 : 84 c4 a2 02 8e 83 c4 20  68
c548 : 4d c6 ce 83 c4 30 12 ce  d0
c550 : 84 c4 ce 84 c4 ad 85 c4  d4
c558 : 38 e9 07 8d 85 c4 4c 47  37
c560 : c5 ad a7 c4 c9 07 d0 0c  ae
c568 : ad a9 c4 d0 07 20 df c5  b1
c570 : 90 c6 b0 1e a2 00 bd a4  be
c578 : c4 f0 03 20 b3 c5 a2 01  6f
c580 : bd a4 c4 f0 03 20 b3 c5  6a
c588 : a2 02 bd a4 c4 f0 03 20  4f
c590 : b3 c5 6c d5 c4 a5 01 09  8b
c598 : 01 85 01 68 85 ff 68 85  ae
c5a0 : fe 68 0d 0d dc 29 01 f0  b4
c5a8 : 04 58 6c d7 c4 68 a8 68  f1
c5b0 : aa 68 40 8e 83 c4 8a 0a  0d
c5b8 : 8d 84 c4 0a 0a 38 ed 83  1b
c5c0 : c4 8d 85 c4 ad 8a c4 85  92
c5c8 : fe ad 8b c4 85 ff ac 84  2c
c5d0 : c4 b1 fe 99 8c c4 c8 b1  d5
c5d8 : fe 99 8c c4 4c 7d c6 ad  86
c5e0 : 88 c4 18 69 03 8d 88 c4  46
c5e8 : 90 03 ee 89 c4 ad 88 c4  4c
c5f0 : 85 fe ad 89 c4 85 ff a0  4b
c5f8 : 00 b1 fe 8d 8a c4 c8 b1  97
c600 : fe f0 31 8d 8b c4 85 ff  69
c608 : ad 8a c4 85 fe a0 00 b1  34
c610 : fe 99 8c c4 c8 c0 06 d0  e3
c618 : f6 a9 00 8d 98 c4 8d 99  ae
c620 : c4 8d 9a c4 8d a7 c4 a9  66
c628 : 01 8d 9b c4 8d 9c c4 8d  5b
c630 : 9d c4 18 60 ad aa c4 f0  67
c638 : 0f ad 86 c4 8d 88 c4 ad  e3
c640 : 87 c4 8d 89 c4 4c ed c5  b0
c648 : 20 16 c7 38 60 ae 83 c4  7f
c650 : bd 98 c4 f0 22 de 98 c4  ad
c658 : f0 01 60 ad a8 c4 d0 0b  a1
c660 : bd 9e c4 29 fe ae 85 c4  c8
c668 : 9d 04 d4 ae 83 c4 bd 95  93
c670 : c4 f0 0a 9d 9b c4 60 de  02
c678 : 9b c4 f0 01 60 ae 83 c4  e5
c680 : a9 00 9d a4 c4 20 04 c5  0e
c688 : c9 00 d0 12 ae 83 c4 ad  3d
c690 : a7 c4 1d a1 c4 8d a7 c4  f6
c698 : a9 01 9d a4 c4 60 10 15  77
c6a0 : c9 f8 90 29 29 07 0a aa  77
c6a8 : bd c3 c4 85 fe bd c4 c4  a3
c6b0 : 85 ff 6c fe 00 c9 61 b0  65
c6b8 : 09 ae 83 c4 9d 92 c4 4c  ac
c6c0 : 7d c6 e9 61 ae 83 c4 9d  9c
c6c8 : 95 c4 4c 7d c6 a8 ae 83  f5
c6d0 : c4 bd 92 c4 9d 98 c4 98  93
c6d8 : 29 0f c9 0c 90 01 60 0a  23
c6e0 : aa 98 4a 4a 4a 4a a8 bd  c7
c6e8 : ac c4 85 fe bd ab c4 c0  05
c6f0 : 0e f0 06 46 fe 6a c8 d0  c9
c6f8 : f6 6c d3 c4 ae 85 c4 9d  17
c700 : 00 d4 a5 fe 9d 01 d4 ae  46
c708 : 83 c4 bd 9e c4 09 01 ae  26
c710 : 85 c4 9d 04 d4 60 a9 00  d6
c718 : 8d 0f dc a9 02 8d 0d dc  14
c720 : ad 9e c4 8d 00 d4 8d 07  ea
c728 : d4 8d 0e d4 ad d7 c4 8d  a9
c730 : 14 03 ad d8 c4 8d 15 03  5f
c738 : 60 78 a5 fe 48 a5 ff 48  60
c740 : ad 86 c4 8d 88 c4 ad 87  88
c748 : c4 8d 89 c4 20 ed c5 90  77
c750 : 02 b0 1e a9 08 a2 49 8d  3d
c758 : 05 d4 8e 06 d4 8d 0c d4  bf
c760 : 8e 0d d4 8d 13 d4 8e 14  96
c768 : d4 a9 0f 8d 18 d4 20 d9  e3
c770 : c4 68 85 ff 68 85 fe 58  29
c778 : 60 ff 00 ff 00 ff 00 00  d8
```

Listing 1. Der Sequenzer. Bitte mit dem MSE eingeben.

```
programm : musik               c800 c950
-------------------------------------
c800 : 15 c8 00 31 c8 00 31 c8  82
c808 : 00 a4 c8 00 a4 c8 00 e9  f1
c810 : c8 00 00 00 00 1b c8 21  17
c818 : c8 27 c8 60 61 ef 48 ef  49
c820 : 00 60 61 ef 48 ef 00 0c  c3
c828 : 6d 93 9a 93 8a 93 9a 93  4f
c830 : 00 37 c8 69 c8 9d c8 18  f8
c838 : 61 ef 0c ba 24 ef 0c ba  53
c840 : 24 ef 0c 6d ba b6 b6 ba  be
c848 : b9 b5 b5 b9 18 61 b9 48  84
c850 : ef 0c c3 24 ef 0c c3 24  71
c858 : ef 0c 6d c3 bb bb c3 c2  4f
c860 : ba ba c2 18 61 c2 30 ef  f8
c868 : 00 0c 61 b3 b5 b6 b5 b3  8c
c870 : b5 b6 b5 b3 b5 6d b6 b3  6d
c878 : b3 b6 b5 b0 b0 b5 18 61  e6
c880 : b5 30 ef 0c b8 ba bb ba  90
c888 : b8 ba bb ba b8 ba 6d bb  72
c890 : b8 b8 bb ba b5 b5 ba 18  0f
c898 : 61 ba 30 ef 00 0c 6d 8a  8c
c8a0 : 93 9a 93 00 aa c8 b8 c8  cb
c8a8 : de c8 18 61 ef 79 d1 d1  d2
c8b0 : c8 c8 c6 c6 c1 61 c1 00  95
c8b8 : 0c 61 c6 c8 ca c8 c6 c8  df
c8c0 : 18 ca 0c c1 c3 c5 c3 c1  76
c8c8 : c3 18 c5 0c bb c1 [illegible] c1  e7
c8d0 : bb c1 18 c3 0c b6 b8 ba  b9
c8d8 : b8 b6 b8 18 ba 00 18 61  eb
c8e0 : ef 48 79 b6 b1 ab 61 a6  74
c8e8 : 00 ef c8 08 c9 2c c9 18  68
c8f0 : 61 ef 60 ef ef ef ef 06  a9
c8f8 : 67 9a a5 aa aa b5 ba ba  24
c900 : c5 18 61 ca ef ef ef 00  c1
c908 : 0c 61 a5 a6 a8 a6 a5 ef  39
c910 : a6 a5 a3 ef a5 a3 a1 ef  4d
c918 : a3 a1 a0 ef 60 ef 0c a3  af
c920 : a1 9b 3c ef 60 95 18 b5  1a
c928 : ef ef ef 00 0c 61 95 96  5a
c930 : 98 96 95 ef 96 95 93 ef  bb
c938 : 95 93 91 ef 93 91 90 ef  e1
c940 : 60 ef 0c 93 91 8b 3c ef  54
c948 : 60 8a 18 aa ef ef ef 00  87
```

Listing 2. Ein Musik-Demo. Bitte mit dem MSE eingeben.

C 64 extern — Der Weg nach draußen (Teil 3)

Nachdem wir in der letzten Folge die Programmierung der Control-Ports abgeschlossen haben, wenden wir uns heute einer der vielseitigsten Schnittstellen des C 64/VC 20 zu — dem User-Port.

Der User-Port führte neben dem Expansion-Port lange Zeit ein Schattendasein, denn an ihm können keine Programm-Module verwendet werden. Mittlerweile hat der User-Port aber weit aufgeholt. Er wird für Steuerzwecke, zum Anschluß eines Druckers, für die RS232 und auch zum Programmieren von EPROMs mit einem Zusatzgerät verwendet. Sehen wir uns diese interessante Schnittstelle etwas genauer an. Die Anschlußbelegung der User-Ports von C 64 und VC 20 zeigen die Bilder 1 und 2. Beachten Sie bitte, daß die Anschlüsse an der Ober- und Unterseite des User-Ports verschiedene Funktionen haben. Klemmen Sie deshalb dort niemals eine Krokodilklemme

1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	H	J	K	L	M	N

PIN	BELEGUNG	PIN	BELEGUNG
1	GND	A	GND
2	+5V, max. 100 mA	B	FLAG2
3	RESET	C	PB0
4	CNT1	D	PB1
5	SP 1	E	PB2
6	CNT2	F	PB3
7	SP 2	H	PB4
8	PC2	J	PB5
9	SER. ATN IN	K	PB6
10	9V AC, max. 100 mA	L	PB7
11	9V AC, max. 100mA	M	PA2
12	GND	N	GND

Bild 1. Anschlußbelegung des C 64 User-Ports. (Bei Aufsicht auf die Computerrückseite)



oder vergleichbares an! Ich möchte Ihnen nun zuerst zeigen, wie der 8-Bit-Parallelport am User-Port programmiert wird. Dieser Port bietet sehr vielseitige Anwendungsmöglichkeiten, obwohl er einfach zu programmieren ist. Angesprochen wird er über die acht Leitungen PB0 bis PB7 (Pin C bis L).

Über den Parallelport kann man Daten einlesen und Daten ausgeben. Das Praktische an dieser Tatsache ist, daß man die Richtung der Daten für jede der acht Leitungen einzeln bestimmen kann. Warum es aber gerade acht Leitungen gibt, hat einen einfachen Grund: Zur Steuerung des Ports gibt es im Computer zwei verschiedene Speicherstellen. Jeder Leitung am Port ist in jeder dieser Speicherstellen genau ein Bit zugeordnet. Daher also acht Leitungen, gleich acht Bit, gleich ein Byte.

Welche Bedeutungen haben nun diese beiden Speicherzellen? Die erste nennt sich »DDR«. Das heißt »Data Direction Register« (Datenrichtungsregister). Hier wird, wie der Name schon sagt, die Datenrichtung festgelegt. Aber wie? Dem Bit X im DDR ist die Datenleitung PBX zugeordnet. Ist zum Beispiel Bit 2 in diesem Register auf 1, dann ist die Leitung PB2 als Ausgang deklariert. Eine 0 macht die entsprechende Leitung zum Eingang. So einfach ist das!

Die zweite bereits genannte Speicherstelle enthält die eigentlichen Daten. Man bezeichnet sie als Portregister.

Nehmen wir einmal an, wir hätten in das Datenrichtungsregister die Zahl 255 eingeschrieben, also alle Leitungen als Ausgänge programmiert. Eine 255 im Portregister legt dann High-Pegel auf alle acht Datenleitungen. Eine 0 im Portregister bewirkt GND auf allen Datenleitungen. Man kann also wie im Datenrichtungsregister bitweise den Spannungszustand am Ausgang bestimmen (1=high, 0=low). Hierbei ist Bit 0 wieder PB0 zugeordnet. Die Spannungspegel, die wir am Port erhalten, können wir dann »extern« verarbeiten, doch davon später.

Im zweiten Beispiel benutzen wir den Parallelport als Eingang. Nehmen wir an, wir hätten in das Datenrichtungsregister eine 0 eingeschoben, also alle Datenleitungen als Eingänge definiert. Jetzt müssen wir »externe« Daten an den Port anlegen. Das geschieht durch Verbinden der Datenleitungen entweder mit +5 V oder GND der Computerversorgungsspannung. Dabei entspricht einer »0« GND und eine »1« +5V. Eine »1« entsteht auch, wenn eine Leitung nicht angeschlossen, also weder mit +5 V noch mit GND verbunden wird. Das spart oft viel Schaltungsaufwand. Die an den Port angelegte Bitkombination kann jederzeit durch Abfragen des Portregisters ausgelesen werden. Der neue Zustand der Datenleitungen wird, wenn sich etwas geändert hat, jeweils sofort und automatisch in das Portregister übernommen.

Die beiden Funktionen Eingänge/Ausgänge dürfen, wie schon gesagt, beliebig gemischt werden. So ist beispielsweise möglich, die Leitungen 2, 5 und 7 als Ausgänge und alle restlichen Leitungen als Eingänge zu definieren. Dazu müßte man im Datenrichtungsregister die Zahl 164 ablegen (164 = Bits 2, 5 und 7 gesetzt). Die Information über die Spannungszustände an den Ausgängen müßte in die Bits 2, 5 und 7 des Portregisters gePOKEt und die Information über die Eingänge aus den Bits 0, 1, 3, 4 und 6 des Portregisters ausgelesen werden.

Nach soviel grauer Theorie wollen wir endlich wieder etwas programmieren. Bevor es losgeht, müssen Sie noch wissen, wie die Adressen der eben besprochenen Speicherstellen lauten. Sie sind in Tabelle 1 aufgelistet. Jetzt haben wir alles nötige Rüstzeug, um voll in eine sinnvolle Anwendung des Computers einzusteigen: Programmieren wir eine Alarmanlage!

1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	H	J	K	L	M	N

PIN	BELEGUNG	PIN	BELEGUNG
1	GND	A	GND
2	+5V, max. 100 mA	B	CB1
3	RESET	C	PB0
4	JOY 0	D	PB1
5	JOY 1	E	PB2
6	JOY 2	F	PB3
7	LIGHT PEN	H	PB4
8	CASSETTE SWITCH	J	PB5
9	SERIAL ATN IN	K	PB6
10	9V AC, max. 100 mA	L	PB7
11	9V AC, max. 100 mA	M	CB2
12	GND	N	GND

Bild 2. Anschlußbelegung des VC 20 User-Ports. (Bei Aufsicht auf die Computerrückseite)

Alarmanlage per User-Port

Tippen Sie Listing 1 ab und starten Sie das Programm mit RUN. Auf dem Bildschirm erscheint der Kontaktplan des Parallelports. Hoffentlich haben Sie beim Einkauf für die letzte Folge an den Stecker für den User-Port gedacht, den brauchen wir nämlich jetzt.

Bringen Sie an einem der vier Eckkontakte (= GND, siehe Bild 1) des User-Port-Steckers einen Draht an, und stecken Sie den Stecker in den User-Port. Zum Anbringen des Drahtes ist Löten wohl das Beste. Keinesfalls dürfen Sie einen anderen Anschluß versehentlich berühren, beziehungsweise mit GND in Verbindung bringen.

Nehmen Sie sich nun das andere Ende des Drahtes zur Hand, und berühren Sie damit eine der Datenleitungen PB1 bis PB7. Bitte nicht die Datenleitung PB0, da diese vom Programm als Ausgang (Alarmanzeiger) benutzt wird (was wir mit diesem machen können, sehen wir in der nächsten Folge). Welche Pins die richtigen sind, können Sie Bild 1 und 2 entnehmen.

Beim Berühren erklingt ein Alarmton, der Bildschirmrahmen ändert die Farbe, und der Alarmanzeiger (PB0) geht auf high (+5 V). Zusätzlich wird auf dem Bildschirm angezeigt, welche Kontakte den Alarm ausgelöst haben. Wenn Sie keine der Datenleitung mehr mit dem Draht berühren, die Eingänge also alle High-Signal erhalten, werden alle Alarmsignale sofort gelöscht.

Werfen wir einen Blick auf das Listing. Nach der Initialisierung der Variablen, der Portregister, des SID-Chips und des Bildschirms wird in Zeile 390 das Portregister (Variable R) so lange ausgelesen, bis ein anderer Wert als 254 auftaucht. Warum 254? 254 ist die Summe der Wertigkeiten der Bits 1 bis 7, also aller Bits des Portregisters ohne den Alarmanzeiger. Das ist der Ruhezustand der Alarmanzeige; alle Kontakte sind geöffnet (= High-Pegel).

Wird ein Alarm ausgelöst, dann werden ab Zeile 430 sämtliche Alarmanzeiger eingeschaltet. Ab Zeile 470 wird ein neuer Kontaktplan ausgegeben. Dabei wird jeder Kontakt auf Alarm überprüft und das Ergebnis angezeigt.

Anschließend wartet der Computer auf das Alarmende. Wenn noch Alarm besteht, wird wieder ein neuer Kontaktplan ausgegeben, weil sich am Kontaktzustand etwas geändert haben kann. Andernfalls werden die Alarmanzeiger ausgeschaltet und es wird zur Bildschirminitialisierung zurückgesprungen.

Wer sich das Programm genau ansieht, wird schnell feststellen, daß es so, wie es hier abgedruckt ist, zum realen Einsatz als Alarmanlage weniger geeignet ist. Dazu hat es verschiedene Schwächen. Zum einen wird

	DATEN-RICHTUNGS-REGISTER	PORTREGISTER
VC 20	37138	37136
C 64	56579	56577

Tabelle 1. Zusammenstellung der User-Port-Register

	DATEN-RICHTUNGS-REGISTER	PORTREGISTER
C 64, PORT 1	56323	56321
C 64, PORT 2	56322	56320
VC 20	37139 / 37154	37137 / 37152

Tabelle 2. Zusammenstellung der Control-Port-Register

	ANSCHLUSS	PIN	BIT	DATENRICHTUNGSREGISTER	PORTREGISTER
RECHTS	JOY 3	4	3	56323	56321
LINKS	JOY 2	3	2	56323	56321

Tabelle 3. Daten zur Abfrage der Paddle-Knöpfe beim Port 1 des C 64

	ANSCHLUSS	PIN	BIT	DATENRICHTUNGSREGISTER	PORTREGISTER
RECHTS	JOY 3	4	3	56322	56320
LINKS	JOY 2	3	2	56322	56320

Tabelle 4. Daten zur Abfrage der Paddle-Knöpfe beim Port 2 des C 64

nicht mit Sicherheit jeder Alarm erkannt: Wenn der Alarmimpuls sehr kurz ist, kann es passieren, daß der C 64 gerade etwas anderes tut, als das Portregister auszulesen, zum Beispiel den Rücksprung nach 390 ausführen. Außerdem ist der eigentlich alarmauslösende Zustand am Portregister nicht der, der ausgewertet wird, weil der auszuwertende Zustand erst nach dem Alarmauslösen in Zeile 420 ermittelt wird. In der Zwischenzeit kann sich schon etwas verändert haben. Das genannte trifft aber nur auf sehr kurze Impulse zu.

Zum anderen ist die Auswertung des Alarmes viel zu einfach gehalten, eben nur als Beispiel. Wenn hier ein Alarm von einer Sekunde auftritt, wird auch nur eine Sekunde lang Alarm gegeben, was natürlich viel zu wenig ist.

Codeschloß

Kommen wir zu einem weiteren Programmbeispiel, dem Codeschloß in Listing 2. Wir haben es wieder mit Raumsicherung zu tun. Wieder wird die Datenleitung PB0 als Ausgang und die restlichen Datenleitungen als Eingänge benutzt.

Was leistet das Programm? Sie müssen durch Verbinden der Datenleitungen PB2 bis PB7 einen sechsstelligen Code eingeben. Wenn Sie keinen Fehler gemacht haben, wird die Datenleitung PB0 auf high gelegt, also beispielsweise eine Tür geöffnet oder eine geheime Anlage eingeschaltet. Wie Sie solche Dinge ansteuern können, erfahren Sie im nächsten Teil dieses Kurses. Über die Leitung PB1 wird das Codeschloß zurückgesetzt (sowohl nach dem Auslösen des Schlosses als auch nach einer Fehlereingabe). Jede Berührung an den Eingängen wird durch einen Signalton quittiert.

In den Zeilen 250/260 werden zwei Variablen deklariert, deren Bedeutung ich kurz erläutern möchte. CL enthält die Länge des einzugebenden Codes, C1$ den Code selbst. Den Inhalt dieser beiden Variablen können Sie nach Belieben verändern. Der von mir gewählte Code ist bewußt sehr primitiv.

In C2$ wird der Code aufgebaut, den Sie durch Berühren

1	ANSCHLUSS	PIN	BIT	DATENRICHTUNGSREGISTER	PORTREGISTER
RECHTS	JOY 3	4	7	37154	37152
LINKS	JOY 2	3	4	37139	37137

Tabelle 5. Daten zur Abfrage der Paddle-Knöpfe beim VC 20

der »externen« Kontakte eingeben. Der große Vorteil hierbei ist, daß die Eingabeeinheit für den Code an einer anderen Stelle (zum Beispiel im Nebenraum) als der Computer sein kann. Man wird also von der Computertastatur unabhängig. Mit einem langen Code wird ein Knacken theoretisch unmöglich. Da sich das Programm durch REM-Anweisungen selbst dokumentiert, bleibt nur noch zu sagen, daß es der Einfachheit halber bei jedem »RESET« an Kontakt PB1 einfach neu mit RUN gestartet wird.

Auch bei diesem Programm ist nur eine Minimallösung angegeben. Man kann es um viele Raffinessen erweitern, so daß zum Beispiel ein Einbruchsversuch als solcher gedeutet wird und so weiter.

Nachdem wir nun wissen was sich am User-Port abspielt, wollen wir noch einmal einen Schritt zurückgehen. Im ersten Teil dieses Kurses sind ein paar Fragen offen geblieben, die ich jetzt beantworten möchte.

Da war als erstes die Sache mit der Umschaltung des Computers auf Joystickeingabe. Vielleicht haben Sie schon eine Ähnlichkeit zwischen Joystick- und Parallelport-Programmierung entdeckt. Bei beiden mußte etwas umgeschaltet werden. Dann gab es zwei mögliche Zustände: 0 oder 1! Die Umschaltung des Computers auf Joystickeingabe ist nichts anderes als die Programmierung des zugehörigen Datenrichtungsregisters. Bitte erinnern Sie sich: Wir haben bei der Umschaltung die zugehörigen Bits gelöscht, also die Datenleitungen der Control-Ports zu Eingängen gemacht. Wer darüber nachdenkt, wird schnell zu der Frage kommen, was denn passiert, wenn wir in die DDRs einfach Einsen hineinschreiben. Die Antwort ist trivial: Die Joystickleitungen werden zu Ausgängen! Doch darum wollen wir uns erst im dritten Teil dieses Kurses kümmern. Die Zusammenstellung aller Adressen, die hierfür interessant sind, zeigt Tabelle 2.

Als nächstes interessiert Sie wahrscheinlich, warum eigentlich Paddles, Joystick, Lichtgriffel und Tastatur einander behindern. Das kommt daher, daß alle diese Eingabeeinheiten dieselben Datenleitungen benutzen. Aus diesem Grund sind auch die Umschaltungen notwendig.

Warum aber benutzt die Tastatur Leitungen am Control-Port? Eine Computertastatur ist durch eine Matrix aus Zeilen und Spalten mit dem Computer verbunden. An jedem Kreuzungspunkt sitzt eine Taste, die, wenn sie gedrückt wird, die beiden Leitungen des Kreuzungspunktes miteinander verbindet. Die Routine des Betriebssystems, welche die Tastatur abfragt, legt nun an eine Zeile ein Signal an und überprüft, in welcher Spalte dieses Signal auftritt. Daraus läßt sich ableiten, welche Taste gedrückt ist. Wird kein Signal entdeckt, so ist in dieser Zeile keine Taste gedrückt, und das Signal wird an die nächste Zeile angelegt. Aus diesem Verfahren erklärt sich, warum einige Tasten Priorität gegenüber anderen haben. Das können Sie feststellen, indem Sie mehrmals zwei Tasten gleichzeitig drücken. Es wird immer dieselbe Taste vom Computer registriert werden.

Diese Zeilen- und Spaltenmatrix liegt auf denselben Leitungen wie der Joystick, weil auch hier »externe« Signale gegeben und Informationen gelesen werden müssen, wozu unser Computer eben nur eine bestimmte Anzahl von Bausteinen hat. Deshalb funktioniert die Tastatur nach dem POKEn in die Control-Port-DDRs meist nicht mehr, weil sie nicht mehr ordnungsgemäß decodiert wird. Die Werte, die in bestimmte Speicherzellen immer wieder hineingeschrieben werden (siehe Folge 1) stammen auch aus der Tastaturdecodierungs-Routine.

Die Störung der Tastatur kommt nun daher, daß es dem Computer völlig gleichgültig ist, ob Sie den Joystickhebel bewegen oder auf die Tasten hämmern. Er prüft artig seine Spalten nach und kümmert sich nicht darum, woher die festgestellte Verbindung denn nun kam. Die Störung durch den Lichtgriffel hat dieselbe Ursache.

Jetzt müssen wir nur noch einen Rückstand aus der letzten Folge aufklären. Dabei handelt es sich um die Feuerknöpfe der Paddles. Ich kann Ihnen erst jetzt zeigen wie sie abgefragt werden, weil wir auch hier die Kenntnisse über die Programmierung des Parallelports benötigen. Es funktioniert nämlich schon wieder genauso: Man setze im zugehörigen DDR die entsprechenden Bits auf Null (Leitung auf Eingang schalten) und frage dann die entsprechenden Bits im zugehörigen Portregister ab. Natürlich zeigt auch hier wieder eine Null im Portregister an, daß der Feuerknopf am Paddle gedrückt wurde. In Tabelle 3 sind die nötigen Informationen für Port 1 des C 64, in Tabelle 4 die für den Port 2 und in Tabelle 5 die für den VC 20 zusammengefaßt. Dabei sind die Paddles gemäß der Position ihres Verbindungskabels am Control-Port mit links und rechts bezeichnet. Das rechte Paddle ist also an POT X, das linke an POT Y angeschlossen. Da die Feuerknöpfe der Paddles dieselben Leitungen wie der Joystick benutzen, gilt hier ebenfalls das oben Beschriebene über die gleichzeitige Benutzung von Paddles und Tastatur.

Damit wollen wir es für heute bewenden lassen. Das nächstemal sehen wir dann, wie versprochen, wo sich den Control-Ports Signale entnehmen lassen. Außerdem wollen wir uns mit Anschlußmöglichkeiten verschiedener Adapter an den User-Port beschäftigen und unter anderem einen 220 Volt-Adapter für unseren Computer bauen!

(Tobias Nicol/aw)

```
100 REM ********************************   <238>
110 REM *                              *   <159>
120 REM *    A L A R M A N L A G E     *   <233>
130 REM *    ---------------------     *   <226>
140 REM *                              *   <189>
150 REM *       BY TOBIAS NICOL        *   <199>
160 REM *                              *   <209>
170 REM *     NEUWIESENSTRASSE 20      *   <172>
180 REM *                              *   <229>
190 REM *      6000 FRANKFURT 71       *   <012>
200 REM *                              *   <249>
210 REM ********************************   <092>
220 :                                      <196>
230 REM ******* INITIALISIERUNG *******    <201>
240 R = 56577 : POKE 56579,1 : POKE R,0    <219>
250 :                                      <226>
260 REM ***** SID-CHIP EINSTELLEN *****    <076>
270 T = 54296 : SI = 54272                 <202>
280 POKE SI+ 7,207 : POKE SI+ 8,200        <131>
290 POKE SI+13,240 : POKE SI+11,17         <002>
300 :                                      <022>
310 REM ***** BILDSCHIRM AUFBAUEN *****    <083>
320 PRINT "{CLR,2DOWN,5RIGHT}ALARMANLAGE"  <119>
330 PRINT "{5RIGHT}==========={3DOWN}"     <003>
340 FOR A = 1 TO 7                         <123>
350 PRINT"KONTAKT"A": O. K.{DOWN}"         <029>
360 NEXT A                                 <126>
370 :                                      <092>
380 REM *********** ALARM? ************    <253>
390 IF ((PEEK(R))AND 254) = 254 THEN 390   <089>
400 :                                      <122>
410 REM *********** ALARM! ************    <043>
420 IR = PEEK (R)                          <060>
430 POKE 53280,2                           <230>
440 POKE R , PEEK ( R ) OR 1               <069>
450 POKE T,15                              <249>
460 :                                      <182>
470 REM **** KONTAKTPLAN AUSGEBEN  ****    <023>
480 PRINT "{HOME,6DOWN}"                   <000>
490 FOR A = 1 TO 7                         <017>
500 IF (IR AND (2↑A)) = 0 THEN 530         <065>
510 PRINT "{12RIGHT}O. K.{SPACE,DOWN}"     <090>
520 GOTO 540                               <036>
530 PRINT "{12RIGHT}ALARM!{DOWN}"          <143>
540 NEXT A                                 <052>
550 :                                      <018>
560 REM ****** IMMER NOCH ALARM? ******    <100>
570 IR = PEEK ( R )                        <212>
580 IF ( IR AND 254 ) <> 254 THEN 470      <171>
590 :                                      <058>
600 REM ** ALARMANZEIGER AUSSCHALTEN **    <063>
610 POKE 53280,254                         <017>
620 POKE R , PEEK ( R ) AND 254            <233>
630 POKE T,0                               <066>
640 :                                      <108>
650 REM ********* RUECKSPRUNG *********    <160>
660 GOTO 310                               <112>
670 :                                      <138>
680 :                                      <148>
690 :                                      <158>
700 REM ********************************   <074>
710 REM *** AENDERUNGEN FUER VC-20  ***    <027>
720 REM ********************************   <094>
730 REM                                    <028>
740 REM LOESCHEN SIE  : ZEILEN 270-290.    <078>
750 REM                                    <048>
760 REM GEBEN SIE EIN :                    <151>
770 REM                                    <070>
780 REM 240 R=37136:POKE37138,1:POKER,0    <076>
790 REM 270 T=36878 : POKE36875,200        <068>
800 REM 430 POKE 36879,26                  <072>
810 REM 610 POKE 36879,27                  <178>
```

Listing 1. Die User-Port-Alarmanlage. Beachten Sie bitte die Eingabehinweise auf Seite 54.

```
100 REM ********************************   <238>
110 REM *                              *   <159>
120 REM *    C O D E S C H L O S S     *   <122>
130 REM *    ---------------------     *   <226>
140 REM *                              *   <189>
150 REM *       BY TOBIAS NICOL        *   <199>
160 REM *                              *   <209>
170 REM *     NEUWIESENSTRASSE 20      *   <172>
180 REM *                              *   <229>
190 REM *      6000 FRANKFURT 71       *   <012>
200 REM *                              *   <249>
210 REM ********************************   <092>
220 REM                                    <026>
230 REM ******* INITIALISIERUNG *******    <201>
240 R = 56577 : POKE 56579,1 : POKE R,0    <219>
250 T = 54296 : I = 1 : CL = 6             <179>
260 C1$ = "123456" : C2$ = ""              <178>
270 REM                                    <078>
280 REM ***** SID-CHIP EINSTELLEN *****    <096>
290 SI = 54272                             <077>
300 POKE SI+ 7,207 : POKE SI+ 8,200        <151>
310 POKE SI+13,240 : POKE SI+11,17         <022>
320 REM                                    <128>
330 REM ***** BILDSCHIRM AUFBAUEN *****    <103>
340 PRINT "{CLR,DOWN,5RIGHT,BLACK}CODESCHL
OSS"                                       <187>
350 PRINT "{5RIGHT}-----------{CYAN}"      <237>
360 PRINT "{2DOWN}GEBEN SIE BITTE DEN{DOWN
}"                                         <059>
370 PRINT "CODE EIN. JEDER KON-{DOWN}"     <078>
380 PRINT "TAKT WIRD MIT EINEM{DOWN}"      <031>
390 PRINT "TON BESTAETIGT."                <050>
400 PRINT "{2DOWN}(RESET: KONTAKT 2)"      <140>
410 REM                                    <218>
420 REM ***** KONTAKT BETAETIGT?  *****    <042>
430 PR = (( PEEK (R) AND 254 ))            <158>
440 IF ( PR AND 254 ) = 254 THEN 420       <083>
450 REM                                    <002>
460 REM *** KONTAKTNUMMER ERRECHNEN ***    <162>
470 X = LOG ( 254 - PR ) / LOG (2) - 1     <103>
480 REM                                    <032>
490 REM ***** WERTE KONTROLLIEREN *****    <205>
500 IF ( X < 0 ) OR ( X > 6 ) THEN 420     <011>
510 IF X <> INT (X) THEN 420               <244>
520 REM                                    <074>
530 REM *********** RESET?  ***********    <084>
540 IF X = 0 THEN RUN                      <111>
550 REM                                    <104>
560 REM * EINZUGEBENDEN CODE AUFBAUEN *    <109>
570 C2$ = C2$ + RIGHT$ ( STR$ (X) , 1 )    <107>
580 REM                                    <134>
590 REM ***** SIGNALTON AUSGEBEN  *****    <195>
600 POKE T,15                              <145>
610 FOR A = 1 TO 30 : NEXT A               <065>
620 POKE T,0                               <056>
630 REM                                    <184>
640 REM **** SIND SCHON CL ZAHLEN  ****    <062>
650 REM ****  EINGEGEBEN WORDEN?   ****    <010>
660 I = I + 1                              <135>
670 IF I = CL + 1 THEN 720                 <005>
680 REM                                    <234>
690 REM ********* RUECKSPRUNG *********    <200>
700 GOTO 420                               <176>
710 REM                                    <008>
720 REM * EINGEGEBENER CODE RICHTIG?  *    <163>
730 IF C1$ = C2$ THEN 790                  <116>
740 REM                                    <038>
750 REM *** CODE IST NICHT RICHTIG! ***    <010>
760 PRINT "{2DOWN,GREEN}FEHLER!!!{CYAN}"   <132>
770 FOR A = 1 TO 2000 : NEXT A : RUN       <239>
780 REM                                    <080>
790 REM ****** CODE IST RICHTIG! ******    <027>
800 PRINT "{2DOWN,GREEN}RICHTIG!!!{CYAN}"  <019>
810 PRINT "{DOWN}BITTE MIT KONTAKT 2"      <141>
820 PRINT "{DOWN}LOESCHEN!"                <116>
830 REM                                    <130>
840 REM *** USER-PORT-SIGNAL SETZEN ***    <081>
850 POKE R , ( PEEK (R) ) OR 1             <179>
860 REM                                    <160>
870 REM **** AUF DEN RESET WARTEN  ****    <066>
880 IF (( PEEK (R)) AND 2) = 2 THEN 870    <105>
890 RUN                                    <170>
900 REM                                    <200>
910 REM                                    <210>
920 REM                                    <220>
930  REM *******************************   <222>
940  REM *** AENDERUNGEN FUER VC-20 ***    <003>
950  REM *******************************   <242>
960  REM                                   <004>
970  REM LOESCHEN SIE  : ZEILEN 290-310    <183>
980  REM                                   <024>
990  REM GEBEN SIE EIN :                   <127>
1000 REM                                   <044>
1010 REM 240 R=37136:POKE 37138,1          <205>
1020 REM 250 POKE R,0:T=36878:I=1:CL=6     <000>
1030 REM 290 POKE 36876,220                <229>
```


Listing 2. Das Codeschloß

Memory Map mit Wandervorschlägen (11)

Bei unserer Wanderung durch die Speicherlandschaft treffen wir heute auf die Speicherstellen 183 bis 199. Sie sind unter anderem auch für die Filenummer, Geräteadressen und Sekundär-Adresse verantwortlich.

Gelegentlich erwähne ich bei meinen Erklärungen, daß wichtige Dinge der Commodore-Computer nicht in den Handbüchern stehen.

Kürzlich erhielt ich zu diesem Thema den Brief eines Lesers, der mich darauf hinwies, daß entgegen meiner Behauptung die meisten der von mir vermißten Erklärungen in einem Handbuch, nämlich im »64 Intern« von Data Becker zu finden sind. Natürlich hat dieser Leser recht: In diesem Buch steht in der Tat sehr viel Informatives. Ich benutze es selbst oft und kann es den fortgeschrittenen Computer-Amateuren sehr empfehlen.

Nur, lieber Leser, ich halte es nicht für ein Handbuch, sondern für Fachliteratur. Wenn ich Handbuch sage, dann meine ich die von Commodore offiziell herausgegebenen Schriften — für Anfänger und Amateure. Und diese Gattung könnte wirklich etwas ergiebiger sein.

Adresse 183 ($B7)

Länge des derzeitigen Filenames

Die LOAD-, SAVE- und VERIFY-Befehle für Disketten verlangen die Angabe eines Programm- oder Dateinamens, auf Computerdeutsch »File-Name«. Nähere Angaben dazu finden Sie im Texteinschub # 1 »Files-Geräte-Namen-Nummern«.

Auch der OPEN-Befehl kann einen File-Name haben. Bei Kassettenoperationen kann der File-Name weggelassen werden.

In der Speicherzelle 183 steht während und nach der Verwendung eines der oben genannten Befehle eine Zahl, die angibt, aus wievielen Zeichen der File-Name besteht.

Bei Disketten sind File-Namen möglich, die aus maximal 16 Zeichen bestehen.

Bei Kassetten dagegen sind Namenslängen von maximal 187 Zeichen erlaubt. Allerdings werden vom Computer auf dem Bildschirm nur 16 Zeichen ausgedruckt (siehe dazu den Texteinschub »Tape-Header«).

Für die Längenangabe in Zelle 183 gilt dabei nur die Anzahl derjenigen Zeichen, die zwischen den Gänsefüßchen stehen.

Diese Zahl kann nach einer Ein-/Ausgabeoperation, auch nach einer ungültigen oder abgebrochenen, durch PEEK (183) ausgelesen werden.

Ein File-Name wird übrigens auch bei einem OPEN-Befehl der RS232-Schnittstelle angegeben. Dieser Name, der bis zu vier Zeichen lang sein kann, wird in die Speicherzellen 659 bis 662 übertragen und gibt dort die Übertragungsrate, Wortlänge und Parity-Prüfung an.

Adresse 184 ($B8)

Nummer der derzeitigen Datei (File)

Hinter jedem OPEN-Befehl steht eine Zahl, die der durch diesen Befehl angefangenen Datei zugeordnet wird. Diese Datei- oder File-Nummer gilt als Referenz für alle anderen Ein- und Ausgabebefehle derselben Datei. Nähere Angaben dazu können Sie dem nebenstehenden Texteinschub Nr. 1 »Files-Geräte-Namen-Nummern« entnehmen.

Ein OPEN-Befehl ruft die entsprechende Routine des Betriebssystems auf, welche die File-Nummer in die Speicherzelle 184 schreibt. Vor dort kann sie mit PEEK(184) ausgelesen werden. Geben Sie die folgende Zeile direkt ein:

```
A=30:OPEN A,3:PRINT PEEK
(184):CLOSE A
```

Um verschiedene File-Nummern auszuprobieren, definieren wir sie als Variable A. Nach dem »A« des OPEN-Befehls steht die Zahl 3. Damit wird der Bildschirm angewählt (siehe »Sekundär-Adresse« im schon erwähnten Texteinschub). Das Anwählen des Bildschirms vermeidet eine störende Meldung des Betriebssystems.

Mit RETURN nach der obenstehenden Zeile wird der jeweilige Wert von A als Inhalt der Zelle 184 ausgedruckt.

Adresse 185 ($B9)

Derzeitige Sekundär-Adresse

Die Sekundär-Adresse steht als dritte Angabe hinter den Ein- und Ausgabe-Befehlen LOAD, SAVE, VERIFY und OPEN. Sie hat bei den verschiedenen Peripheriegeräten spezielle Funktionen. Diese Funktionen sind im nebenstehenden Texteinschub näher erläutert.

Der jeweilige Wert der Sekundär-Adresse steht in der Speicherzelle 185, allerdings um 96 erhöht. Für Sekundär-Adressen stehen, über die Standardwerte der einzelnen Peripheriegeräte hinaus, die Zahlen von 0 bis 31 zur Verfügung. Ab 32 fängt in Zelle 185 wieder der Zyklus ab 0 an. Das können wir uns anschauen. Ich wähle zur Eröffnung einer Datei wieder den Bildschirm als »nicht-störendes« Empfangsgerät.

```
A=15:OPEN 1,3,A:PRINT
PEEK(185)—96:CLOSE 1
```

Durch Verändern des Wertes von A können Sie alle Möglichkeiten durchspielen.

Adresse 186 ($BA)

Derzeitige Geräte-Nummer

Jedes an den Computer anschließbare Gerät hat eine eigene Nummer, die zusammen mit den Ein-/Ausgabe-Befehlen LOAD, SAVE, VERIFY und OPEN angegeben werden muß. Wird keine Nummer angegeben, nimmt der Computer automatisch an, daß die Datasette gemeint ist.

Alle von Commodore vorgegebenen Geräte-Nummern sind in der folgenden Tabelle aufgelistet.

Geräte-Nummer	angesprochenes Gerät
0	Tastatur
1	Datasette
2	RS232- (User-Port) Schnittstelle
3	Bildschirm
4	Drucker (normal)
5	Drucker (zusätzlich)
8	Disketten-Laufwerk Nr. 0
9	Disketten-Laufwerk Nr. 1
10, 11	weitere Disketten-Laufwerke

Die normale Geräte-Nummer eines Druckers ist 4, die eines Disketten-Laufwerks 8. Die zusätzlichen Nummern müssen gesondert am betreffenden Gerät eingestellt werden.

Nach der Ausführung eines der oben genannten Befehle steht die entsprechende Geräte-Nummer in der Speicherzelle 186, aus der sie mit PEEK(186) ausgelesen werden kann.

Adresse 187 bis 188 ($BB bis $BC)

Zeiger auf Adresse des derzeitigen File-Namens

Die Bedeutung eines Programm- oder Dateinamens — normalerweise kurz »Files-Name« genannt, sind im nebenstehenden Texteinschub »File-Geräte-Namen-Nummern« näher beschrieben. In den Speicherzellen 187/188 steht in der Low/High-Byte-Darstellung ein Zeiger auf diejenige Adresse im Programm-Speicher, wo dieser Name gespeichert ist.

Eine Ausnahme ist hier der OPEN-Befehl der RS232-Schnittstelle. Ihr File-Name wird in die Speicherzellen 659 bis 662 gebracht, wo er verschiedene Parameter dieser Schnittstelle steuert.

Adresse 189 ($BD)

Zwischenspeicher für RS232-Parity-Prüfung und für Kassettenoperationen

Die RS232-Routinen benutzen diese Speicherzellen als Zwischenspeicher für ein Prüf-Byte (Parity-Prüfung) bei der Ausgabe. Die Parity-Prüfung habe ich kurz im letzten Heft im neunten Teil des Kurses erklärt.

Auch die Kassetten-Routinen bedienen sich dieser Speicherzelle. Sie verwenden sie als Zwischenspeicher für das gerade gesendete oder empfangene Zeichen.

Adresse 190 ($BE)

Blockzähler für Kassetten-Ein-/Ausgabe

Das Betriebssystem des Computers schreibt bei SAVE ein Programm zweimal auf das Band der Datasette. Beim LOAD-Befehl wird der erste Block in den Arbeitsspeicher des Computers geladen; der zweite — identische — Block wird dann mit dem ersten Block Byte für Byte verglichen, um Datenfehler auf dem nicht immer ganz zuverlässigen Bandmaterial zu erkennen.

In der Speicherzelle 190 wird dem Betriebssystem angezeigt, wieviele Blockteile bei diesem Prozeß noch gelesen oder gespeichert werden müssen. Vom Basic-Programm aus ist diese Speicherzelle nicht zugänglich.

Adresse 191 ($BF)

Zwischenspeicher für LOAD-Operationen vom Band

Diese Speicherzelle wird beim Laden eines Programms vom Band dazu benutzt, um Zeichen aus einzelnen Bits zusammenzusetzen.

Adresse 192 ($C0)

Motorsperre der Datasette

Die Tasten der Datasette werden sechzigmal in der Sekunde von der »Interrupt-Routine« des Betriebssystem überprüft, ob eine von ihnen gedrückt worden ist. Die Speicherzelle 192 spielt dabei eine entscheidende Rolle, beim C 64 allerdings in einer anderen Weise als beim VC 20. Wie sie diese Rolle beim C 64 spielt, ist im Zusammenhang mit der Speicherstelle 1 ganz am Anfang dieses Kurses beschrieben worden (siehe 64'er-Ausgabe, 11/84, Seite 173). Ich habe dabei in zwei Beispielen gezeigt, wie durch Abfrage des vierten Bits von Adresse 1 geprüft werden kann, ob eine Taste der Datasette gedrückt ist und wie der Motor durch Setzen und Löschen des Bit 5 der Zelle 1 ein- und ausgeschaltet werden kann. Vorausgesetzt, der Inhalt der Speicherzelle 192 ist ungleich Null und eine Taste der Datasette ist gedrückt.

Heute will ich, wie damals versprochen, denselben Vorgang für den VC 20 beschreiben.

Wie Sie sich vielleicht noch erinnern, wird die Speicherzelle 1 beim VC 20 nicht für die Steuerung der Ein- und Ausgänge des Mikroprozessors verwendet. Diese Rolle wird beim VC 20 durch zwei Register des »Versatile Interface Adapter« (VIA 6522-A) ausgefüllt.

Für die Abfrage der Datasetten-Tasten ist das sechste Bit des VIA-Registers 37151 zuständig. Bei gedrückter Taste steht es auf 1, sonst auf 0. Ein kleines Programm zeigt es Ihnen:

```
10  X = PEEK(37151)
20  PRINT X
30  IF X = 62 THEN 50
40  GOTO 10
50  PRINT"TASTE
    GEDRÜCKT"
```

Wenn keine Taste gedrückt ist, läuft ein Zahlenband mit 126 ab. Die entsprechende Darstellung als Dualzahl lautet 1111 1110. Bei einer gedrückten Taste steht in 37151 die Zahl 62, als Dualzahl 0011 1110. Wichtig, wie gesagt ist nur das zweithöchste Bit.

Mit der Abfrage der Zeile 30 springt beim Drücken einer Taste das Programm auf die Zeile 50 und druckt den Text aus.

Den Motor der Datasette können wir mit Hilfe des Registers 37184 schalten. Wie beim C 64 gilt auch jetzt, daß dazu die hier angesprochene Speicherzelle 192, auch Interlock-Register genannt, eine Zahl größer als 0 enthält und daß außerdem eine Taste der Datasette gedrückt ist. Drücken Sie auf PLAY und geben Sie direkt ein:

POKE 192,255
POKE 37148,251: der Motor bleibt stehen.
POKE 37148,252: der Motor läuft los.

Bestimmend sind hier Bit 2, 3 und 4.

Zum Ausschalten muß lediglich Bit 2 auf 1 stehen, zum Einschalten die drei Bits auf 110. Jede Zahl, die als Dualzahl diese Bedingungen erfüllt, kann dafür hergenommen werden. Um unabhängig von den anderen Bits des Registers 37148 zu bleiben, die ja auch ganz bestimmte andere Funktionen haben, empfiehlt es sich, über Boole'sche Verknüpfungen nur die wichtigen drei Bits zu verändern. Die beiden POKE-Befehle sehen dann so aus:

Ausschalten: POKE 37148,PEEK (37148) OR 2
Einschalten: POKE 37148,PEEK (37148) AND 12

Adresse 193 bis 194 ($C1 bis $C2)

Anfangsadresse für Ein-/Ausgabe-Operationen

In diesen Speicherzellen steht in Low/High-Byte-Darstellung die Adresse, ab der ein Programm gerade geladen oder gespeichert wird. Dieses Adresse wird übrigens von hier auch in die Speicherzellen 172/173 gebracht, die wir schon früher besprochen haben.

Bei LOAD und SAVE auf Band steht hier die Anfangsadresse des Bandpuffers (828). Im Bandpuffer steht allerdings nur der sogenannte Bandvorspann (auf englisch »Tape Header«), während der Hauptteil des Programms im Programmspeicher ab einer Adresse steht, auf die der Zeiger in den Speicherzellen 195/196 hinweist.

Adresse 195 bis 196 ($C3 bis $C4)

Zeiger auf den Anfang des Programms hinter dem Tape Header

Bei jedem LOAD- und SAVE-Befehl für Kassetten wird der Vorspann (Tape Header), in dem Programmtyp, Anfangs- und Endadresse aufgezeichnet sind, im Kassettenpuffer ab Adresse 828 gespeichert. Der eigentliche Teil des Programms steht dann im Programmspeicher.

In den Speicherzellen 195/196 steht in der Low-High-Byte-Darstellung diese Adresse, ab der das Programm beginnt. Ich habe für alle diejenigen, die mit der Datasette arbeiten, im nebenstehenden Texteinschub # 2 »Tape-Header« die Zusammenhänge mit einem Beispiel dargestellt.

Adresse 197 ($C5)

Tasten-Code der zuletzt gedrückten Taste

In Ausgabe 6/85 des 64'er auf Seite 123 habe ich Ihnen mit Wort und Bild beschrieben, wie die Tasten des Computers abgefragt werden. Die dabei für jede Taste entstehende Dualzahl wird in eine Dezimalzahl (0 bis 63) umgewandelt und zuerst in die Speicherzellen 203 beziehungsweise 653 gebracht. Zur Umwandlung und Abfrage der Zellen 203 und 653 bringe ich das nächste Mal mehr Details. Nach der Prüfung, welche Taste gedrückt worden ist, wird die Codezahl von 203 in die Speicherzelle 197 gebracht und dort »aufgehoben«. Diese vermeintliche Verdoppelung wird vom Betriebssystem dafür gebraucht, um zu erkennen, ob die nächste gedrückte Taste mit der vorhergehenden identisch ist. Ist sie identisch, dann entscheidet der Inhalt der Speicherzelle 650, ob das Zeichen dieser Taste mehrfach ausgedruckt wird. In 650 steht die sogenannte Wiederholfunktion. Aber ich will nicht vorgreifen. Die Codezahlen der einzelnen Tasten werde ich bei der Besprechung der Zelle 203 auflisten.

Adresse 198 ($C6)

Anzahl der Zeichen im Tastaturpuffer

Die Funktion des Tastaturpuffers, zu dem wir bei den Speicherzellen 631 und 640 noch kommen werden, habe ich bereits in diesem Kurs und zwar in Ausgabe 7/85 auf Seite 141/142 im Texteinschub »Dynamische Tastenabfrage« erklärt. Dabei habe ich damals schon sozusagen im Vorgriff die Zelle 198 verwendet.

In dieser Speicherzelle steht die jeweilige Anzahl der Zeichen, die im Tastaturpuffer gespeichert sind und darauf warten, weiterverarbeitet zu werden.

Das folgende kleine Programm zeigt es.

```
10  GET A$
20  PRINT PEEK (198);A$
30  FOR J=1 3000:NEXT J
40  GOTO 10
```

Der GET-Befehl holt ein Zeichen aus dem Tastaturpuffer — sofern eines dort zu finden ist. Die Zeile 20 druckt die Anzahl der Zeichen im Puffer aus, daneben das erste dieser Zeichen. Dann folgt eine Warteschleife, die uns erlaubt, ganz schnell ein paar Tasten zu drücken. Danach springt das Programm an den Anfang zurück und arbeitet diese eingegebenen Zeichen ab. Es ist dabei deutlich zu sehen, wie durch den GET-Befehl bereits ein Zeichen aus dem Puffer genommen und dadurch der Inhalt der Zelle 198 sofort um 1 reduziert wird.

Der Inhalt der Speicherzelle 198 kann mit POKE auch verändert werden.

Eine sinnvolle Anwendung dieser Beeinflussung erlaubt der nicht gerade sehr populäre WAIT-Befehl.

Ersetzen Sie bitte im obigen Programm die Warteschleife der Zeile 30 durch:

30 POKE 198,0: WAIT 198,1

Zuerst wird dem Computer vorgegaukelt, daß der Tastaturpuffer leer sei. Durch den WAIT-Befehl wartet das Programm danach so lange, bis ein Zeichen im Tastaturpuffer erscheint und springt erst dann auf die nächste Zeile 40.

Wenn Sie nach dem WAIT-Befehl statt der 1 eine 2 eingeben, wartet diese Zeile entsprechend auf zwei Tasteneingaben. Allerdings wird in der Zeile 20 dann nur jedes zweite Zeichen ausgedruckt.

Adresse 199 ($C7)

Flagge für inverse Darstellung der Zeichen

Mit dieser nützlichen Adresse fahren wir das nächste Mal fort.

(Dr. H. Hauck/ah)

Texteinschub #1

Files — Geräte — Namen — Nummern

In den Handbüchern von Commodore und auch in anderen Beschreibungen wird von den Ein- und Ausgabe-Befehlen, wie zum Beispiel LOAD, SAVE, OPEN etc. leider ein recht verwirrendes Bild geboten. Ich beziehe mich dabei auf die hinter diesen Befehlen stehenden Ziffern und Namen.

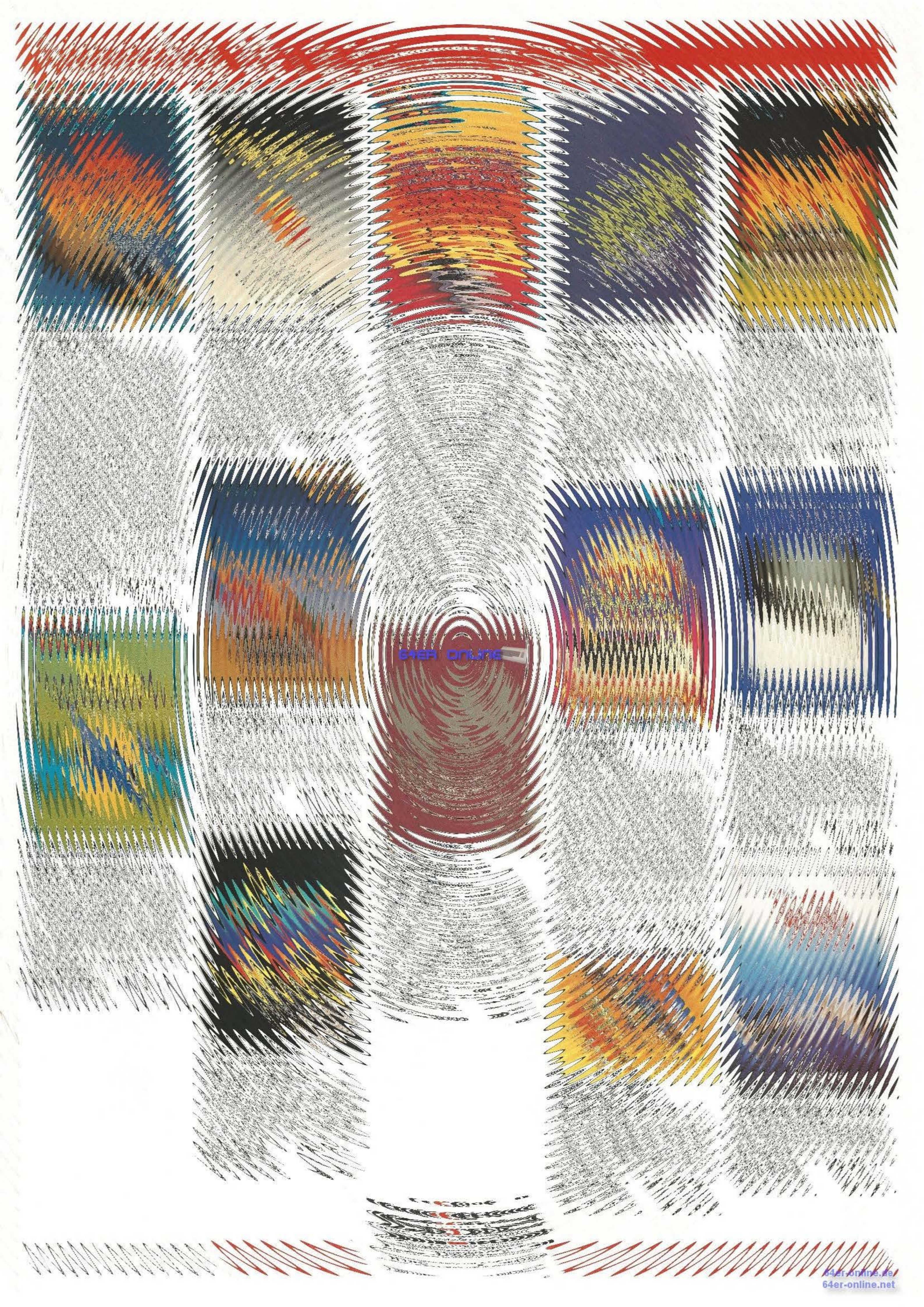
In der Bezeichnung »File-Name« und »Geräte-Nummer« sind sich die Autoren noch weitgehend einig. Daß die dritte Angabe hinter den Befehlen aber Sekundär-Adresse, Command, Speicheradressen-Flag oder gar EOT-Flag genannt wird, muß den Computeramateur zwangsläufig verwirren.

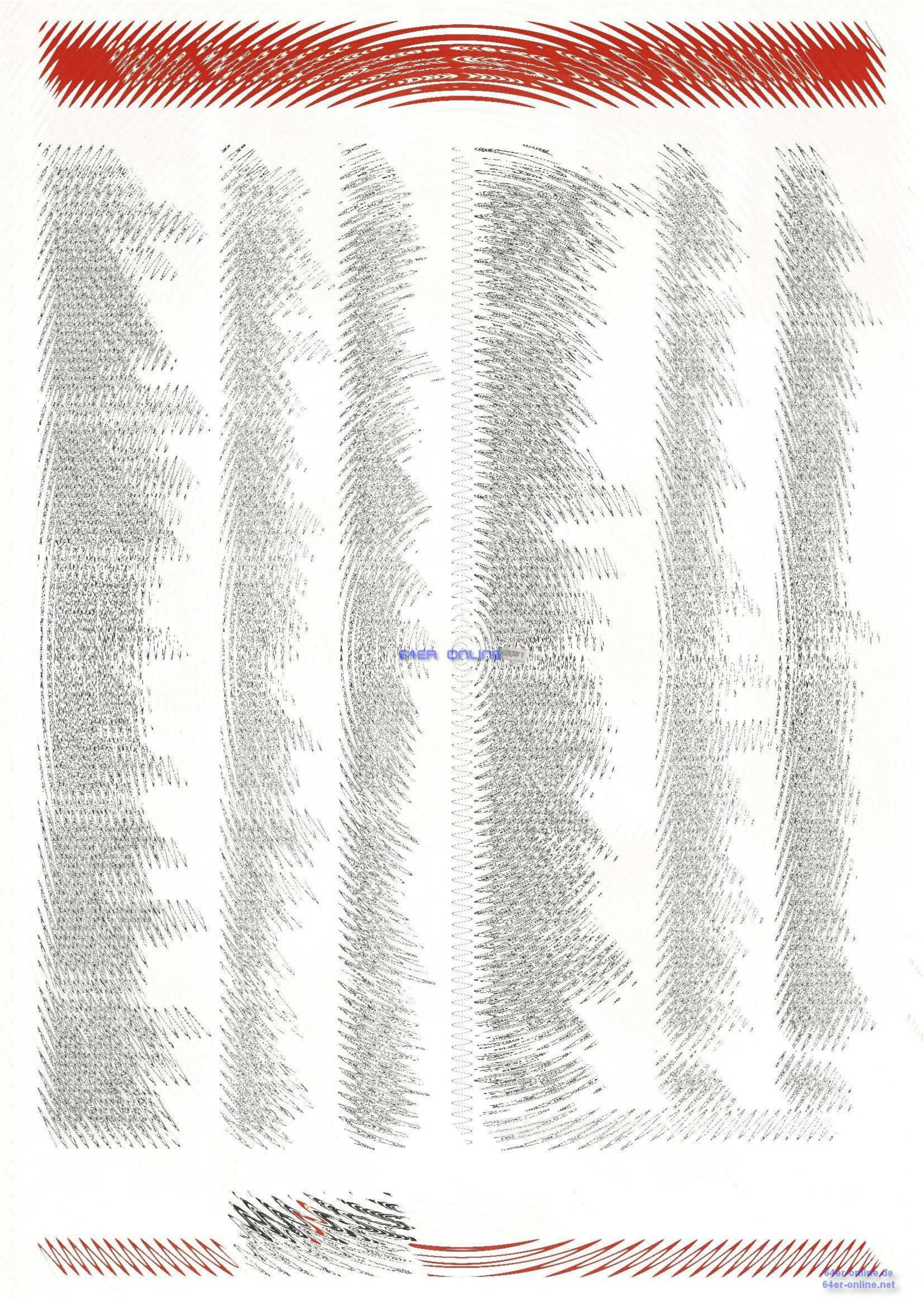
Da wir im nebenstehenden Kurs gerade die Speicherzellen 183 bis 188 besprechen, die alle mit diesen Anhängseln der eingangs genannten Befehle zusammenhängen, ist dies eine gute

Fortsetzung auf Seite 139






64ER ONLINE
64er-online.de
64er-online.net



Fortsetzung von Seite 134

Gelegenheit, etwas Systematik in die Angelegenheit zu bringen.

Als erstes gebe ich die Bezeichnungen der Befehle aus einem Commodore-Buch wieder, bevor die einzelnen Angaben im Detail diskutiert werden.

— LOAD "File-Name", Geräte-Nr., Speicheradressen-Flag
— SAVE "File-Name", Geräte-Nr., EOT-Flag
— VERIFY "File-Name", Geräte-Nr., Speicheradressen-Flag
— OPEN File-Nr., Geräte-Nr., Sekundär-Adresse, "File-Name, Typ, Modus"
— INPUT# File-Nr., Variable
— GET# File-Nr., Variable
— CMD File-Nr.
— CLOSE File-Nr.

File-Name

»File« wird normalerweise mit »Datei« übersetzt.

Einen Datei-Namen gibt es aber nur beim OPEN-Befehl. Bei den Befehlen LOAD, SAVE und VERIFY ist der Name des Programms gemeint, der bekanntlich in Gänsefüßchen hinter diesen Befehlen steht. Bei Disketten als Pflicht, bei Kassetten als Option. Beim OPEN-Befehl steht der Name zwar auch in Gänsefüßchen, aber nicht direkt hinter dem Befehl, sondern erst an vierter Stelle. Auch er kann bei Kassetten-Betrieb, aber auch beim Drucker weggelassen werden.

Bei Disketten-Dateien steht hinter dem Namen — als sein Bestandteil — noch Typ und Modus. Typ bezeichnet die Art der Datei (RELative, SEQuential, USeR, PRoGramm), Modus die Operation (Read, Write, Append).

Der File-Name kann bei Disketten auch eine Anweisung an das Betriebssystem des Disketten-Laufwerks sein (NEW, SCRATCH, RENAME etc.) Nähere Erklärungen dazu finden Sie im Handbuch des Disketten-Laufwerks.

Da in 187/188 die Adresse angegeben ist, ab der der Programm- beziehungsweise Datei-Name im Speicher steht, können wir ihn dort ansehen.

Geben Sie direkt ein
LOAD "1234"

Nach der RETURN-Taste kommt die übliche Aufforderung der Datasette, die wir aber mit der STOP-Taste abwürgen. In 187/188 steht jetzt die Anfangsadresse »A« des gespeicherten Namens:
A = PEEK(187)+256*PEEK(188):PRINT PEEK(A)

Wir erhalten die Zahl 49, das ist der ASCII-Codewert der Ziffer »1«. Durch Erhöhen der Anfangsadresse »A« im PRINT-Befehl um jeweils 1, kommen auch die übrigen Zeichen des Namens zum Vorschein.

Die VC 20-Besitzer kennen sicher den Effekt, der entsteht, wenn beim SAVE-Befehl vor dem File-Namen, aber innerhalb der Gänsefüßchen, eine Farbtaste eingegeben wird. Die FOUND-Meldung druckt nämlich den File-Namen in der gewählten Farbe aus, was besonders bei Kassetten-Operationen sehr eindrucksvoll ist.

Das Disketten-Laufwerk bietet eine andere Überraschung. Die geSHIFTete SPACE-Taste innerhalb des File-Namens, erzeugt das zweite Gänsefüßchen.
SAVE"SPIEL 2 (SHIFT-SPACE) SYS 16000",8
erscheint in der Directory als: "SPIEL 2" SYS 16000 und kann mit LOAD"SPIEL 2",8 geladen werden.

Geräte-Nummer

Alle an den Computer anschließbaren Peripheriegeräte haben eine eigene Nummer, unter welcher sie »angesprochen« werden können. Es handelt sich eigentlich um eine Adresse, und in der Tat sprechen manche Autoren von der »Primär-Adresse«. In der Beschreibung der Speicherzelle 184 sind sie alle zusammengestellt.

Bei den Befehlen LOAD, SAVE, VERIFY und OPEN steht diese Geräte-Nummer als zweite Angabe, vor der ersten Angabe durch ein Komma getrennt:
LOAD"NAME",1.....bedeutet "Laden von Datasette"
LOAD"NAME",8.....bedeutet "Laden von Diskette"
OPEN 12,4.....bedeutet "An den Drucker"
(Die 12 hinter dem OPEN-Befehl ist eine »File-Nummer«. Sie kommt weiter unten an die Reihe.)

Die Gerätenummer kann auch weggelassen werden. Dann allerdings nimmt der Computer automatisch an, daß es sich um die Datasette handelt. Die Geräte-Nummer steht jeweils in der Speicherzelle 186 und kann von dort mit PEEK ausgelesen werden.

Sekundär-Adresse

Die dritte Angabe hinter LOAD, SAVE und OPEN hat widersprüchliche Namen, wohl deshalb, weil sie abhängig von der Gerätenummer verschiedene Aufgaben hat. Von den ganz am Anfang schon genannten Varianten gefällt mir »Sekundär-Adresse« am besten.

Die folgende Tabelle faßt die Funktionen der Sekundär-Adresse zusammen.

Sie sehen, die Sekundär-Adressen haben es in sich! Die gerade benutzte Sekundär-Adresse steht in der Speicherzelle 185 und kann mit PEEK von dort ausgelesen werden.

File-Nummer

Die File-Nummer, oft auch logische File-Nummer genannt, steht als erste Angabe hinter dem OPEN-Befehl und den damit verbundenen PRINT#-, INPUT#-, GET#- und CLOSE-Befehlen.

Sie gibt einer zu bearbeitenden Datei eine Nummer, die von den nachfolgenden, anderen Befehlen ebenfalls verwendet werden muß, wenn sie sich auf dieselbe Datei beziehen. Auf diese Weise können mehrere Dateien nebeneinander bearbeitet werden, und zwar fünf bei Disketten und auf alle Geräte verteilt insgesamt zehn.

File-Nummern dürfen die Werte von 1 bis 255 haben. Bei Werten über 127 wird bei einem PRINT#-Befehl nach jedem RETURN-Zeichen — CHR$(13) — zusätzlich ein ASCII-Code für Zeilenvorschub — CHR$(10) — an das Gerät gegeben. Diese Eigenschaft kann bei denjenigen Geräten nützlich sein, die normalerweise auf CHR$(13) ohne Zeilenvorschub reagieren (kein automatisches Line-Feed).

Die File-Nummer steht jeweils in der Speicherstelle 184, von wo sie mit PEEK ausgelesen werden kann. Zum Beispiel, um nachzuprüfen, welches Gerät als letztes angesprochen worden ist.

Abschließend möchte ich nochmals darauf hinweisen, daß in den Speicherzellen 183 bis 188 immer die gerade als letzte verwendete Angabe steht. Wir haben aber gesehen, daß der Computer sich maximal zehn File-Nummern mit dazugehörenden Geräte-Nummern, Sekundär-Adressen etc. merken kann.

Er tut dies in speziellen Tabellen, die in den Speicherzellen 601 bis 630 stehen.

Wir werden uns also noch einmal mit dieser Sache befassen, wenn wir bei dem entsprechenden Adressen angekommen sind.

Befehl	Sekundär Adresse	Funktion
LOAD	0	lädt Programm an den Anfang des Programmspeichers
	1	lädt ein Programm absolut, also an die Adresse, von der ab es gespeichert wurde.
SAVE	0	Normales SAVE, Programm wird bei späterem LOAD an den Anfang des Programm-Speichers geladen (Basic-Programme)
	1	erzwingt bei späterem LOAD des Programms die Speicherung ab der Adresse, wo es zur Zeit steht (Maschinen-Programme)
	2	setzt am Programmende auf der Kassette eine »Band-Ende«-Markierung, die beim »Überlesen« das Band mit der Fehlermeldung »DEVICE NOT PRESENT« stoppt.
	3	Kombination von 1 und 2
OPEN bei Kassette	0	Daten lesen
	1	Daten schreiben
	2	Daten schreiben mit »Bandende«-Markierung
OPEN bei Diskette	0	vom Betriebssystem der Floppy für »Laden« reserviert
bei Diskette	1	vom Betriebssystem der Floppy für »Speichern« reserviert
	2-14	reserviert numerierten Daten-Kanal, bis zu drei gleichzeitig
	15	reserviert Kommando-Kanal (nähere Angaben zu diesen Sekundär-Adressen siehe Floppy-Handbuch)
OPEN bei Drucker	0-10	die Funktionen sind bei den Druckern zum Teil verschieden. Bitte in Drucker-Anleitung nachsehen.

Texteinschub # 2

Tape Header

Wenn ein Programm oder eine Datei auf Band gespeichert wird, setzt der Computer vor das Programm einen Vorspann, der auf englisch »Tape-Header« genannt wird. Da dieser Name weit verbreitet ist, will ich ihn hier beibehalten. Der Tape-Header ist 192 Byte lang. Er enthält alle wichtigen Angaben über das nachfolgende Programm.

Beim Laden eines Programms wird der Tape Header im Kassettenpuffer gespeichert, für den die Speicherstellen 828 bis 1019 reserviert sind. Von dort kann der Inhalt des Tape-Headers gelesen und analysiert werden.

Bevor wir das versuchen, will ich erst seine Zusammensetzung erklären.

Im **ersten Byte** steht eine Kennzahl für den Typ des Programms. Diese Kennzahl ist abhängig von der Sekundär-Adresse, die beim SAVEn eingegeben worden ist. Die Arten der Sekundär-Adressen und ihre Bedeutung ist im anderen Texteinschub »Files-Geräte-Namen-Nummern« genau beschrieben. Es gibt zwei Kennzahlen: 1 und 3.

In Anlehnung an die Erklärung der Sekundär-Adresse kann man die Kennzahl generell dadurch beschreiben, daß ein Programm mit Kennzahl 1 immer an den Anfang des zur Verfügung stehenden Programm-Speichers geladen wird. Hauptsächlich kommt das für Basic-Programme in Frage.

Eine Kennzahl 3 bewirkt, daß das Programm an diejenige Stelle des Programmspeichers geladen wird, wo es vor dem SAVEn gestanden hat. Das ist hauptsächlich der Fall bei Programmen in Maschinensprache.

In Verbindung mit der Bedeutung der Sekundär-Adresse kann man den Zusammenhang wie Tabelle 1 zeigt darstellen:

Sekundär Adresse	Kennzahl	Bedeutung
0 oder leer	1	Basic-Programm
1	3	Maschinen-Programm
2	1	Basic-Programm mit End-Of-Tape-Marke
3	3	Maschinen-Programm mit End-Of-Tape-Marke

Tabelle 1

In **Byte 2 und 3** steht in Low/High-Darstellung die Adresse, ab der das Programm im Speicher des Computers stand, als es gespeichert wurde.

In **Byte 4 und 5** steht die entsprechende End-Adresse des Programms.

Ab **Byte 6 bis Byte 192** steht der Name des Programms. Er darf also maximal 187 Zeichen lang sein. Bei LOAD werden allerdings nur 16 Zeichen auf dem Bildschirm dargestellt.

Jetzt wollen wir das alles mit einem kleinen Experiment überprüfen.

Schreiben Sie bitte ein kleines Programm, es braucht nicht sehr sinnvoll zu sein, wie zum Beispiel:

```
10 REM TAPE HEADER
20 REM TEST PROGRAMM
```

Nehmen Sie ein leeres Band und laden das Programm mit einem Namen, der länger sein soll als 16 Zeichen, zum Beispiel:

```
SAVE "TEST PROGRAMM FUER INHALT TAPE HEADER"
```

Nach Drücken der RECORD- und PLAY-Tasten der Datasette meldet der Computer:

```
FOUND TEST PROGRAMM FU
```

Es werden also nur 16 Zeichen inklusive Leerzeichen gedruckt. Sobald das Programm geladen ist, schauen wir im Kassettenpuffer nach, was in den ersten fünf Bytes steht, danach lesen wir die restlichen Bytes des Puffers.

Geben Sie direkt, ohne Zeilennummer, ein:

```
FOR I=0 TO 4: PRINT PEEK (828+I);: NEXT,
```

Sie erhalten die Zahlen 1 1 8 41 8 (beim VC 20 mit 3-K-Speichererweiterung 1 1 4 41 4)

Danach geben wir wiederum direkt ein:

```
FOR I=5 TO 192: PRINT CHR$(PEEK(828+I));: NEXT
```

Beim VC 20 geben Sie in der FOR...NEXT-Schleife eine kleine Zahl ein, da der Bildschirmspeicher beim VC 20 kleiner ist als beim C 64.

Jetzt erscheint der volle Programmname, gefolgt von nicht sichtbaren Leerstellen. Wenn Sie in der letzten Direkteingabe den CHR$-Teil weglassen, dann druckt die Zeile die ASCII-Codes aus, und Sie sehen dann die Leerstellen als Zahl 32.

Diese Resultate habe ich zur besseren Übersicht in Tabelle 2 dargestellt.

Die Kennzahl in Byte »1« können Sie dadurch verändern, daß Sie dem oben verwendeten SAVE-Direktbefehl nach dem langen Namen ein ,1,1 anhängen. Im Ausdruck steht dann die Kennzahl »3«.

Übrigens, wenn Sie in den Speicherzellen 195/196 nachschauen, finden Sie dort denselben Wert wie in den Zellen 829/830, so wie die Beschreibung es in der Memory-Map erklärt.

Vielleicht fragen Sie jetzt nach dem Nutzen dieser ausführlichen Erklärung. Nun, hauptsächlich kann man damit Programme, die eigentlich wegen LOAD ERROR nicht mehr ladbar sind, retten. Oder aber man kann durch Verändern der Zahlen in den Bytes 2 bis 5 nachträglich die Adressen ändern, in die das Programm geladen wird. Die erste Anwendung werde ich erklären, sobald wir zu den Adressen des Kassetten-Puffers selbst kommen.

Das Problem des LOAD oder SAVE mit geänderten Adressen ist aber zu umfangreich für einen Texteinschub innerhalb dieses Kurses. Es wäre eigentlich einen eigenen kleinen Beitrag wert.

Adresse	828	829	830	831	832	833 etc.
Byte Nr.	1	2	3	4	5	6 etc.
Bedeutung	Kennzahl	Low High Byte		Low High Byte		Namen in ASCII-Code
Resultat	1	1	8	41	8	T etc.
bei C 64	1	(2089)				(2049)
Resultat	1	1	4	41	4	T etc.
bei VC 20		(1065)				(1025)

Tabelle 2

Fortsetzung von Seite 45

möchten, finden den Anschluß BASIC an Pin 20 vom Sockel des Basic-ROMs beziehungsweise der Adapterplatine, und Basic-ROM an Pin 20 des ROMs selber, wo die Leitungen dann anzuschließen sind. Am einfachsten ist dies wohl durch Herausbiegen des Pins am Basic-ROM (Achtung: nicht abbrechen) zu bewerkstelligen, so daß dieser nicht im Sockel steckt und ein Kabel angelötet werden kann.

Die Zusatzplatine ist so ausgelegt, daß man sie mit an den Schrauben, die auch zur Befestigung der Tastatur dienen, installieren kann. Ob man sie in der linken oder rechten Seite anbringt, ist völlig belanglos, nur sollte man die Taster auf der Gegenseite befestigen, beziehungsweise vorher prüfen, ob die Einbautiefe der Taster gering genug ist, so daß diese sich nicht mit der Zusatzplatine berühren.

Die Taster werden an dem oberen Gehäuseteil über der Tastatur montiert, wobei man darauf achten sollte, daß genügend Platz für die ausgewählten Taster zur Verfügung steht (Durchmesser; am besten vom Inneren des Gehäuses anpassen).

Ist die Adapterplatine so aufgesteckt, daß die Kerben der ICs in Richtung Kassettenport zeigen und mit EPROM-Trans, dem Kernal und mindestens einem EPROM 1 bis 3 bestückt, sind wir fertig und können den Computer wieder zusammenbauen.

Eventuell kann die Abschirmung — dies ist die Pappe mit der Alu-Schicht — so aufgeschnitten werden, daß beim Verlegen der Kabel von Adapter zur Zusatzplatine keine Probleme auftreten.

Die Steckverbindung Tastatur — Computer kann nur in einer Richtung aufgesteckt werden; bei der Leuchtdiode zeigt das rote Kabel zum Ein-Aus-Schalter.

Gehäuse zusammenschrauben, fertig.

Ich betreibe diese Schaltung seit etwa 6 Monaten und möchte sie nicht mehr missen. Was ich damit aber zum Ausdruck bringen will, ist, daß die Stromversorgung den zusätzlichen Belastungen gewachsen ist; auch sind bis dato noch keine Wärmeprobleme aufgetreten.

Mir ist bekannt, daß sich viele C64-Besitzer schwer dazu durchringen können, ihren Computer äußerlich oder innerlich (Achtung! Garantieverlust) zu verändern. Diesmal ist es aber, glaube ich, eine Überlegung wert. Alle, die nicht die Möglichkeit zum Herstellen der Platinen haben, können sich an den Verlag wenden. Lesen Sie dazu unseren Hardware-Leserservice. (Peter Rausche/aw)





Assembler ist keine Alchimie — Teil 12

Im Gegensatz zum sonstigen Sprachgebrauch erregt das Kürzel CIA bei Commodore 64-Kennern angenehme Assoziationen. Die beiden CIAs (Complex Interface Adapter) unseres Computers und ihre Rolle bei der Unterbrechungs-Programmierung sollen in dieser Folge entschleiert werden.

Lassen Sie uns kurz rekapitulieren: Als primäre Unterbrechungsanforderer hatten wir drei Bausteine unseres Computers benannt, nämlich den VIC-II-Chip und die beiden CIA-Bausteine. CIA kommt von »Complex Interface Adapter« und ist die Bezeichnung für die beiden Ein- und Ausgabe-Bausteine, die den gesamten Verkehr zwischen dem zentralen Gehirn unseres C 64 und der Peripherie managen. Wir hatten bemerkt, daß ein CIA, der IRQ-CIA (Adressen von 56320 bis 56575), ausschließlich für die maskierbaren Unterbrechungen zuständig ist. Dazu gehören die 60mal pro Sekunde stattfindenden »Timer-Interrupts«, die die Cursorbehandlung, die TI$-Uhr, die Tastaturabfrage etc. bearbeiten. Der andere CIA, genannt NMI-CIA, (Adressenraum 56576-56831) ist nur für die nicht maskierbaren Unterbrechungen verantwortlich und wird bei normaler Nutzung des C 64 so gut wie nie eingesetzt. Ich gehe im folgenden davon aus, daß Sie keine RS232C-Schnittstelle in Ihren Computer eingesetzt haben. Sollte das aber der Fall sein, dann müßten Sie darauf achten, die folgenden Beispiele — die den NMI-CIA betreffen — ohne gleichzeitigen Betrieb dieser Schnittstelle anzuwenden, weil sich sonst Störungen ergeben könnten.

In der Folge 10 dieser Serie (64'er, Ausgabe 7/85) haben wir uns ein Register (das Register 13, Interrupt-Kontrollregister) der CIAs schon genauer angesehen und auch die Unterschiede beider Bausteine festgestellt. Dort war dann die Rede von Timern, Echtzeituhren, Alarm-Funktionen etc. Was es damit auf sich hat und wie man diese Möglichkeiten nutzen kann, das soll diesmal unser Thema sein. Wir werden uns dazu alle Register der CIAs genauer ansehen, die für die von uns ausgewählten Unterbrechungsoptionen eine Rolle spielen. Dabei fallen einige unter den Tisch — das habe ich aber schon in Folge 10 angekündigt —, nämlich diejenigen, die mit dem Verkehr über den seriellen Port, beziehungsweise über die RS232C-Schnittstelle, zu tun haben. Es bleibt dann anderen — kompetenteren — überlassen, darüber zu schreiben.

Register Nr. ($)	Adresse (dez.) CIA-1	CIA-2	Name	Funktion
04	56324	56580	TALO	TIMER A LSB
05	56325	56581	TAHI	TIMER A MSB
06	56326	56582	TBLO	TIMER B LSB
07	56327	56583	TBHI	TIMER B MSB
08	56328	56584	TOD10THS	$^1/_{10}$-Sekunden-Register
09	56329	56585	TODSEC	Sekunden-Register
0A	56330	56586	TODMIN	Minuten-Register
0B	56331	56587	TODHR	Stunden-Register, AM/PM-Flagge
0D	56333	56589	JCR	Unterbrechungs-Kontrollregister
0E	56334	56590	CRA	Kontrollregister A
0F	56335	56591	CRB	Kontrollregister B

Tabelle 1. Die wichtigen Register der beiden CIAs

Wie wäre es zum Beispiel mit Ihnen?

Auch so bleibt uns genug zu tun. In Tabelle 1 finden Sie zunächst eine Übersicht der von uns behandelten Register.

Sie sehen darin, daß jeder CIA über zwei sogenannte Timer (A und B) verfügt, sodann über die »Time of Day« (zu deutsch etwa »Tageszeit«) genannte Echtzeituhr mit vier Registern und schließlich noch über drei Kontrollregister, zu denen auch das schon erwähnte Register 13 gehört. Sehen wir uns zunächst die Timer an.

Die Timer der CIAs.

Insgesamt verfügen wir über vier dieser Timer: Timer A und B im CIA1 und dasselbe nochmal im CIA2. Es handelt sich dabei um 16-Bit-Register, in die ein Startwert geschrieben werden kann, von dem an dann heruntergezählt wird. Jedesmal, wenn dann der Wert 0 unterschritten ist, gibt es für uns die Möglichkeit, bestimmte Ereignisse stattfinden zu lassen. Man kann diese Register unabhängig voneinander, aber auch kombiniert, benutzen. Ein Lesen des Registers liefert immer den momentan gerade aktuellen Wert. Ein Schreiben in das Register führt automatisch zum Festlegen eines Startwertes. Was an Optionen mit diesen Timern möglich ist, wird über Kontrollregister gesteuert. Das CRA (Register $ 0E) bezieht sich vor allem auf den Timer A, das CRB (Register $ 0F) auf Timer B. Die 16-Bit-Register werden — wie gewohnt — in der Form LSB/MSB betrieben. In den Timer A des CIA1 wird bei jedem I/O-Reset folgendes Wertepaar eingetragen:

56324 dezimal 37 LSB
56325 dezimal 64 MSB

Das entspricht einem Startwert von 16421. Im PAL-System hat der Quarz, der die Taktfrequenz bestimmt, eine Frequenz von 17.734472 MHz. Die Prozessorfrequenz errechnet sich daraus mittels Division durch 18 zu 985248.4 Hz (also etwas weniger als 1 MHz, was den europäischen C 64 langsamer macht als den amerikanischen, der etwas mehr als 1 MHz verwendet). Wenn mit dieser Geschwindigkeit der Timer heruntergezählt wird, erhält man genau einen Unterlauf alle $^1/_{60}$ Sekunden. Das ist der Weg, eine kontrollierte Zeitspanne durch den Timer zählen zu lassen. Sei X der gesuchte Startwert, der zu einer Spanne von T Sekunden führt, dann kann man X berechnen mittels:

X = 985248.4 * T

Der Integerwert von X ist dann in ein LSB und ein MSB zu teilen und in die Timer-Register einzutragen. Allerdings ergibt sich so eine natürliche Grenze. Die höchste durch 2 Bytes darstellbare Zahl ist ja 65535. Wenn wir diesen Wert in den Timer schreiben, dann ist er alle $^1/_{15}$ Sekunden auf 0 heruntergezählt. Für längere Zeiten ist aber vorgesorgt. Die beiden Timer A und B sind kombinierbar (wie, dazu kommen wir gleich noch) zu einem 32-Bit-Register. Die höchste Zahl X ist dann:

4 294 967 296 = 2^32

Damit kann im Extremfall eine Herabzählzeit von 1 Stunde, 12 Minuten und zirka 40 Sekunden eingeplant werden, was für die meisten Zwecke ausreichen dürfte.

Möchten Sie also genau eine Sekunde Spielraum haben beim Herunterzählen, dann muß die Zahl 985248 als 4-Byte-Integer-Wert in die Speicher von Timer A und Timer B gebracht werden. Das führt dann zu den Werten 0, 15, 8, 160 (weil 985248 = 0*16777216 + 15*65536 + 8*256 + 160). 0 und 15 gelangen als MSB beziehungsweise LSB in Timer B (also Register 07 und 06), 8 und 160 sind MSB und LSB für den Timer A (Register 05 und 04). Sehen wir uns nun an, wie wir dem Computer sagen, was mit diesen Startwerten in den Timer-Registern geschehen soll. Die beiden Kontrollregister CRA und CRB beziehen sich weitgehend auf die gleichnamigen Timer. Im Bild 1 finden Sie das Register $0E, also CRA und in Bild 2 das andere Kontrollregister CRB ($0F):

Die Bedeutung der Bits 0 bis 4 ist — jeweils für den dazugehörigen Timer — identisch:

Bit 0

an dieser Stelle führt zum sofortigen Anhalten des Timers. 1 in diesem Bit startet das Herunterzählen.

Bits 1 und 2

Diese beiden Bits hängen mit dem externen Signalverkehr zusammen und sollen für uns außer acht bleiben.

Bit 3

Ist dieses Bit = 1, dann ist der sogenannte »One Shot«-Betrieb des Timers aktiv. Das bedeutet, daß vom Startwert an heruntergezählt wird bis auf Null. Es findet nun das programmierte Ereignis statt (zum Beispiel ein IRQ). Anschließend wird der Startwert wieder eingeladen und der Timer gestoppt.

Im Gegensatz dazu läuft der »Continuous«-Betrieb, wenn das Bit den Wert 0 enthält. Dabei geschieht zunächst dasselbe wie beim One Shot Modus, der Timer wird aber nicht angehalten, sondern der ganze Vorgang wiederholt sich in einer Endlosschleife.

Bit 4

Ein Hineinschreiben einer 1 in dieses Bit erzeugt ein sofortiges Neuladen der Timer-Register mit dem Startwert. Dabei ist es gleichgültig, ob der Timer gerade läuft oder nicht. Schreibt man eine Null ein, hat das keine Wirkung.

Beim Lesen des Registers ist dieses Bit immer 0.

Zu diesem Bit und seiner Wirkung ist noch etwas zu sagen. Das Neuladen des Timers geschieht
— immer dann, wenn ein Unterlauf stattgefunden hat oder
— falls der Timer steht und in die Register ein Startwert geschrieben wird. Dabei ist der CIA so konstruiert, daß man kein zwangsweises Laden (also mit Bit 4 = 1) braucht, wenn man den Startwert in der Reihenfolge LSB MSB in die Register bringt.

Die Bits 5 bis 7 haben nun unterschiedliche Bedeutung im CRA und im CRB:

Register CRA ($0E)

Bit 5:

Ist dieses Bit gleich Null, dann wird im Systemtakt gezählt. Den hatten wir vorhin zur Zeitberechnung schon verwendet. Wenn das Bit auf 1 gesetzt ist, zählt der Timer externe Signale.

Bit 6:

Spielt für den Signalverkehr über den seriellen Port eine Rolle und soll uns hier nicht weiter beschäftigen.

Bit 7:

Damit steuert man nicht den Timer A, sondern dieses Bit bezieht sich auf die gleich noch zu behandelnde Echtzeituhr.

Register CRB ($0F)

Die Bits 5 und 6 sind hier im Zusammenhang von Bedeutung. Es gibt vier Kombinationsmöglichkeiten:

Bit 6 — Bit 5

0 — 0

Der Timer B wird — wie vorhin der Timer A — im Systemtakt heruntergezählt.

0 — 1

Der Timer B wird durch externe Signale heruntergezählt.

1 — 0

Der Timer B zählt die Unterläufe von Timer A. Das ist der vorhin erwähnte Punkt, der beide Timer kombiniert zum 32-Bit-Zähler. Man kann also im Extremfall 65536 mal 65536 Takte zählen lassen.

1 — 1

Auch in diesem Fall zählt Timer B die Unterläufe von Timer A. Er tut das aber nur, wenn ein bestimmtes externes Signal vorhanden ist.

Bit 7:

Auch beim Register CRB steuert dieses Bit bestimmte Möglichkeiten der Echtzeituhr. Deshalb haben Sie noch ein wenig Geduld, bis wir diese Uhr behandeln.

Wir kennen uns nun ganz gut aus, wie wir mit den Timern umzugehen haben. Unser Wissen soll in einem kleinen Test erprobt werden. Dazu bedienen wir uns des 1/60 Sekunden IRQ. Wir verändern diese regelmäßige Unterbrechung derart, daß sie nur noch einmal in der Sekunde geschieht. Welche Zahlen dazu in ein 32-Bit-Register gepackt werden müssen, haben wir schon vorhin berechnet. Jeweils in der Reihenfolge LSB/MSB müssen wir sie einschreiben und vorher die Timer anhalten, indem die Bits 0 der Kontrollregister CRA und CRB auf 0 gesetzt werden. Nach dem Einschreiben und Starten der beiden Timer müssen folgende Bitmuster in CRA und CRB stehen:

CRA
Bit 0 = 1 Start Timer A
Bit 3 = 0 Dauerlauf
Bit 5 = 0 Systemtakt

CRB
Bit 0 = 1 Start Timer B
Bit 3 = 0 Dauerlauf
Bit 5 = 0
Bit 6 = 1 Timer B zählt Unterläufe von Timer A.

Bevor wir die Timer starten, muß auch noch das Interrupt-Kontrollregister verändert werden (das hatten wir uns in der 10. Folge genauer angesehen). Bislang erzeugt ein Unterlauf des Timer A eine Unterbrechung. Wir möchten aber, daß der Timer B (damit wir das 32-Bit-Register voll ausnutzen) der Auslöser ist. Dazu muß Bit 0 des ICR gelöscht und statt dessen Bit 1 gesetzt werden.

Im Programm »Timer-Test« (siehe Listing 1 und 2) ist all das realisiert. Mit SYS 49152 gestartet, zeigt sich sofort ein deutlich verlangsamter Cursor. Noch langsamer kann alles werden, indem Sie höhere Werte in die Timer-Register schreiben. Den Normalzustand stellen Sie einfach durch Drücken der RUN/STOP- und der RESTORE-Tasten her. Dabei wird ja — wie Sie aus der letzten Folge her wissen, auch ein I/O-Reset ausgeführt, der den Ausgangszustand wiederherstellt.

Die Verlängerung des IRQ-Zyklus hat übrigens noch einen sinnvollen Nebeneffekt. Je seltener ein laufendes Programm unterbrochen wird, desto schneller wird es mit seinen Jobs fertig. Das kann man immer dann tun — im Extremfall sogar den IRQ ganz ausschalten — wenn man die Möglichkeiten, die der Computer während des normalen IRQ anbietet, nur selten oder aber gar nicht braucht.

Die Echtzeituhren

Wir kennen nun fünf Uhren in unserem Computer: Die vier Timer (jeweils A und B im CIA1 und CIA2), die wir, weil wir die Impulszahlen in Zeiteinheiten umrechnen können, zur Zeitmessung einsetzen könnten und die im Basic verfügbare Uhr TI$, die aber — wie wir nun wissen — lediglich die Umsetzung des Timer A im CIA1 in ein bequemer handhabbares Software-Instrument ist. Zudem ist die Ganggenauigkeit dieser Uhr recht gering. Schon einige Kassettenoperationen genügen, sie völlig aus dem Takt zu bringen.

Um so mehr verwundert es, daß zwei hervorragende Echtzeituhren im Commodore 64 so gut wie nie benutzt werden, ja nicht einmal in irgendeiner Weise softwaremäßig unterstützt werden. Vielleicht ist das ein bißchen zuviel »mehr sein als scheinen«, was Commodore da betreibt, wenn man bedenkt, welche verborgenen Schätze da alle zutage gefördert werden können (man denke nur an die hochauflösende Grafik) bei genauer Untersuchung des Computers.

Jeder der beiden CIAs verfügt über solch eine Uhr, die direkt von der Netzfrequenz getaktet wird. Die Zählung der Zeit geschieht in vier Registern (Register $08 bis $0B), die in Bild 3 gezeigt sind.

Vielleicht fällt Ihnen etwas auf, wenn Sie sich diese vier Bytes mal genauer ansehen: Die Speicherung geschieht in Form von Einer- und Zehnerstellen. Das kann also weder im Binärformat noch als ASCII-Zeichen funktionieren. Hier werden die Ziffern als BCD-Zahlen abgelegt. In der 3. Folge dieser Serie wurde dieses »binary coded decimal«-Format erklärt. Das ist lange her (64'er, Ausgabe 11/84) und soll deshalb hier nochmal vorgestellt werden, damit alle wissen, wovon die Rede ist.

In dieser Zahlendarstellung wird jede Dezimalstelle einer Zahl gesondert in eine Binärzahl umgewandelt. Dann ergibt sich der folgende Zusammenhang:

Binär	Dezimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Das war's! Die anderen möglichen Binärkombinationen (also zum Beispiel 1010 etc.) werden nicht benutzt. Die Zahl 25 beispielsweise lautet im BCD-Format:

```
0010 0101
 ^    ^
 2    5
```

Jetzt ist es Ihnen sicherlich verständlich, warum für die Sekunden- und Minuten-Zehnerstellen nicht mehr als drei Bits reserviert wurden: größer als 6 wird die Zehnerstelle nicht.

Zum Stundenregister TODHR ist aber noch etwas zu sagen: Dort ist nur ein Bit reserviert für die Stunden-Zehnerstelle. Die Uhr läuft nicht bis 24 Uhr, sondern lediglich bis 12 Uhr. Zur Unterscheidung, ob vor- oder nachmittags gemeint ist, dient das Bit 7. Dieses sogenannte AM/PM-Flag ist orientiert an der angelsächsischen Gewohnheit, zum Beispiel für 16 Uhr den Ausdruck 4 PM zu verwenden. PM kommt vom lateinischen »post meridiem«, was übersetzt heißt »nach dem Mittag«, wohingegen AM steht für »ante meridiem«, also »vor dem Mittag«. Meint man nun AM, dann muß diese Flagge auf 0, bei PM aber auf 1 gesetzt sein.

Beim Stellen der Uhren sollte eine Reihenfolge eingehalten werden. Sobald nämlich in das Stundenregister geschrieben wird, hält die Zählung automatisch an. Man kann nun die anderen Werte in die Register schreiben. Den Startschuß liefert das Schreiben in das Register

```
programm : prg.timer-testc000 c051
-----------------------------------------

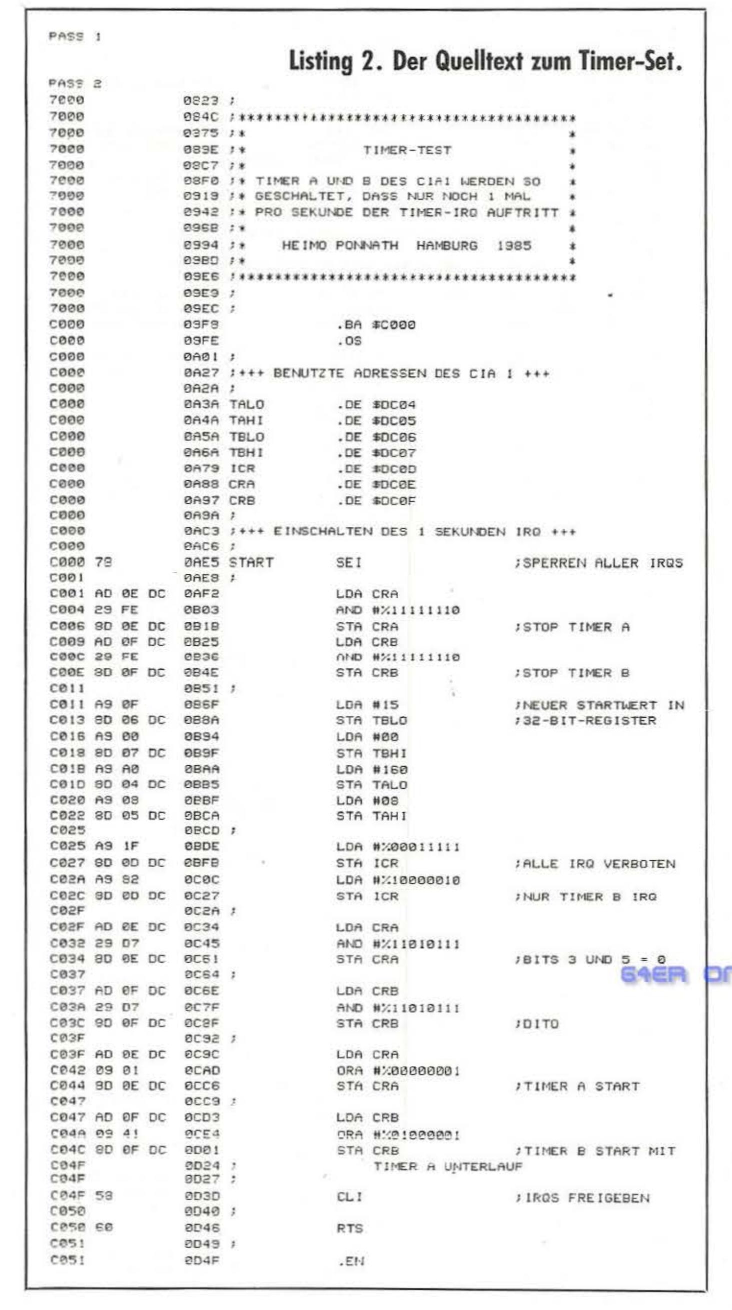
c000 : 78 ad 0e dc 29 fe 8d 0e   4b
c008 : dc ad 0f dc 29 fe 8d 0f   f9
c010 : dc a9 0f 8d 06 dc a9 00   24
c018 : 8d 07 dc a9 a0 8d 04 dc   d5
c020 : a9 08 8d 05 dc a9 1f 8d   84
c028 : 0d dc a9 82 8d 0d dc ad   6e
c030 : 0e dc 29 d7 8d 0e dc ad   0a
c038 : 0f dc 29 d7 8d 0f dc ad   1b
c040 : 0e dc 09 01 8d 0e dc ad   37
c048 : 0f dc 09 41 8d 0f dc 58   a5
c050 : 60 ff 00 ff 00 ff 00 ff   b0
```

Listing 1. Programm Timer-Test, ein Beispiel für die Anwendung eines 32-Bit-Timers

```
programm : obj.alarmuhr c000 c18d
-----------------------------------------

c000 : a9 8e 8d 11 03 a9 c0 8d   11
c008 : 12 03 a9 1d 8d 18 03 a9   a3
c010 : c0 8d 19 03 ad 0e dd 09   12
c018 : 80 8d 0e dd 60 48 8a 48   a1
c020 : 98 48 a9 7f 8d 0d dd ac   49
c028 : 0d dd 10 06 4c 6a c1 4c   a0
c030 : 72 fe 20 bc f6 20 e1 ff   b9
c038 : d0 f5 a2 04 bd 2f fd 9d   b4
c040 : 13 03 ca d0 f7 a2 1a bd   1a
c048 : 35 fd 9d 19 03 ca d0 f7   c0
c050 : a9 7f 8d 0d dc 8d 0d dd   e8
c058 : 8d 00 dc a9 08 8d 0e dc   30
c060 : a9 88 8d 0e dd a9 08 20   fe
c068 : b6 fd 4c 6c fe a9 48 8d   37
c070 : 11 03 a9 b2 8d 12 03 78   2a
c078 : a9 47 8d 18 03 a9 fe 8d   c0
c080 : 19 03 a9 31 8d 14 03 a9   84
c088 : ea 8d 15 03 58 60 24 0d   12
c090 : 30 03 4c 20 c1 20 82 b7   f0
c098 : c0 07 d0 40 ad 0f dd 29   35
c0a0 : 7f 8d 0f dd a0 00 a9 24   5e
c0a8 : 20 fe c0 d0 02 a9 24 c9   23
c0b0 : 13 90 07 f8 38 e9 12 d8   b9
c0b8 : 09 80 8d 0b dd 20 fc c0   1a
c0c0 : 8d 0a dd 20 fc c0 8d 09   ec
c0c8 : dd 20 66 c1 8d 08 dd a9   6b
c0d0 : 00 4c 3c bc 68 68 68 68   d9
c0d8 : a9 ff d0 f5 c0 08 d0 f8   f5
c0e0 : ad 0f dd 09 80 8d 0f dd   1a
c0e8 : a9 84 8d 0d dd a9 3c 85   ff
c0f0 : 04 85 02 a9 ff 85 03 a0   e6
c0f8 : 01 4c a6 c0 a9 60 85 24   dd
c100 : 20 13 c1 0a 0a 0a 0a 85   80
c108 : 25 20 13 c1 05 25 c5 24   13
c110 : b0 c4 60 b1 22 38 e9 30   5d
c118 : 90 ba c9 0a b0 b6 c8 60   5e
c120 : a9 07 20 7d b4 a0 00 ad   b0
c128 : 0b dd 08 29 1f c9 12 d0   73
c130 : 02 a9 00 28 10 05 f8 18   49
c138 : 69 12 d8 20 55 c1 ad 0a   13
c140 : dd 20 55 c1 ad 09 dd 20   96
c148 : 55 c1 ad 08 dd 20 60 c1   ce
c150 : 68 68 4c ca b4 48 4a 4a   a4
c158 : 4a 4a 20 60 c1 68 29 0f   fe
c160 : 09 30 91 62 c8 60 20 13   68
c168 : c1 60 a9 77 8d 14 03 a9   8b
c170 : c1 8d 15 03 4c bc fe c6   d2
c178 : 02 f0 03 4c 31 ea a5 04   46
c180 : 85 02 ad 20 d0 45 03 8d   d4
c188 : 20 d0 4c 31 ea 00 ff 00   f8
```

Listing 3. Eine Echtzeituhr. Bitte beachten Sie die Eingabehinweise auf Seite 54



Listing 2. Der Quelltext zum Timer-Set.

```
PASS 1

PASS 2
7000           0823 ;
7000           084C ;**************************************
7000           0875 ;*                                    *
7000           089E ;*            TIMER-TEST              *
7000           08C7 ;*                                    *
7000           08F0 ;* TIMER A UND B DES CIA1 WERDEN SO   *
7000           0919 ;* GESCHALTET, DASS NUR NOCH 1 MAL    *
7000           0942 ;* PRO SEKUNDE DER TIMER-IRQ AUFTRITT *
7000           096B ;*                                    *
7000           0994 ;*   HEIMO PONNATH  HAMBURG  1985     *
7000           09BD ;*                                    *
7000           09E6 ;**************************************
7000           09E9 ;
7000           09EC ;
C000           09F8            .BA $C000
C000           09FE            .OS
C000           0A01 ;
C000           0A27 ;+++ BENUTZTE ADRESSEN DES CIA 1 +++
C000           0A2A ;
C000           0A3A TALO       .DE $DC04
C000           0A4A TAHI       .DE $DC05
C000           0A5A TBLO       .DE $DC06
C000           0A6A TBHI       .DE $DC07
C000           0A79 ICR        .DE $DC0D
C000           0A88 CRA        .DE $DC0E
C000           0A97 CRB        .DE $DC0F
C000           0A9A ;
C000           0AC3 ;+++ EINSCHALTEN DES 1 SEKUNDEN IRQ +++
C000           0AC6 ;
C000 78        0AE5 START      SEI                ;SPERREN ALLER IRQS
C001           0AE8 ;
C001 AD 0E DC  0AF2            LDA CRA
C004 29 FE     0B03            AND #%11111110
C006 8D 0E DC  0B1B            STA CRA            ;STOP TIMER A
C009 AD 0F DC  0B25            LDA CRB
C00C 29 FE     0B36            AND #%11111110
C00E 8D 0F DC  0B4E            STA CRB            ;STOP TIMER B
C011           0B51 ;
C011 A9 0F     0B6F            LDA #15            ;NEUER STARTWERT IN
C013 8D 06 DC  0B8A            STA TBLO           ;32-BIT-REGISTER
C016 A9 00     0B94            LDA #00
C018 8D 07 DC  0B9F            STA TBHI
C01B A9 A0     0BAA            LDA #160
C01D 8D 04 DC  0BB5            STA TALO
C020 A9 08     0BBF            LDA #08
C022 8D 05 DC  0BCA            STA TAHI
C025           0BCD ;
C025 A9 1F     0BDE            LDA #%00011111
C027 8D 0D DC  0BFB            STA ICR            ;ALLE IRQ VERBOTEN
C02A A9 82     0C0C            LDA #%10000010
C02C 8D 0D DC  0C27            STA ICR            ;NUR TIMER B IRQ
C02F           0C2A ;
C02F AD 0E DC  0C34            LDA CRA
C032 29 D7     0C45            AND #%11010111
C034 8D 0E DC  0C61            STA CRA            ;BITS 3 UND 5 = 0
C037           0C64 ;
C037 AD 0F DC  0C6E            LDA CRB
C03A 29 D7     0C7F            AND #%11010111
C03C 8D 0F DC  0C8F            STA CRB            ;DITO
C03F           0C92 ;
C03F AD 0E DC  0C9C            LDA CRA
C042 09 01     0CAD            ORA #%00000001
C044 8D 0E DC  0CC6            STA CRA            ;TIMER A START
C047           0CC9 ;
C047 AD 0F DC  0CD3            LDA CRB
C04A 09 41     0CE4            ORA #%01000001
C04C 8D 0F DC  0D01            STA CRB            ;TIMER B START MIT
C04F           0D24 ;              TIMER A UNTERLAUF
C04F           0D27 ;
C04F 58        0D3D            CLI                ;IRQS FREIGEBEN
C050           0D40 ;
C050 60        0D46            RTS
C051           0D49 ;
C051           0D4F            .EN
```

7	6	5	4	3	2	1	0
TODIN 50Hz / 60 Hz	externer Signal-verkehr	in Mode	Force-load	ONE Shot / Continu-ous	externer Signal-verkehr		Start / Stop

Bild 1. Das Kontrollregister des Timer A

7	6	5	4	3	2	1	0
ALARM	In MODE		Force-load	ONE Shot / Continu-ous	externer Signal-verkehr		Start / Stop

Bild 2. Dasselbe für den Timer B

Register Name	Nr.	7	6	5	4	3	2	1	0
TODI0THS	08	unbenutzt				1/10-Sekundenwert			
TODSEC	09	unbenutzt	Zehnerstelle Sekunden			Einerstelle Sekunden			
TODMIN	0A	unbenutzt	Zehnerstelle Minuten			Einerstelle Minuten			
TODHR	0B	AM/PM Flagge	unbenutzt		Zehnerstel-le Stunden	Einerstelle Stunden			

Bild 3. Die Register der Echtzeituhren

TODI0TH: von nun an tickt die Uhr wieder.

Ähnlich funktioniert das Lesen der Uhrzeit. Sobald das Stundenregister gelesen wird, führt das zum Anhalten der Uhr, so daß die restlichen Register reibungslos auslesbar sind. Wieder ist es das Zehntelsekundenregister, das beim Auslesen ein Weiterlaufen der Uhr bewirkt. Aber, so werden Sie bemerken, wenn der Auslesevorgang eine bestimmte Zeit beansprucht, führt das zu Verzögerungen? Die Lösung ist, daß der gesamte Inhalt der vier Register gleichzeitig mit dem Auslesen des Stundenwertes in einen internen Speicher transferiert wird und dort weiterläuft. Nach dem Lesen des TODI0TH kommt der aktuelle Wert zurück in die Register und dieser wird weitergezählt.

Nun wird es höchste Zeit, daß wir uns die beiden Bits im CRA und im CRB ansehen, die wir vorhin bei der Timer-Behandlung links liegen ließen. Bit 7 im CRA kündigt der Echtzeituhr an, welche Netzfrequenz zu erwarten ist. Eine 1 an dieser Stelle steht für 50 Hz, eine 0 für 60 Hz. Unser Stromnetz in Deutschland liefert einen Wechselstrom mit 50 Hz, weshalb wir dann dort die 1 setzen sollten. Da gibt es ein kleines Problem: Beim I/O-Reset, der durch Drücken der RUN/STOP- und der RESTORE-Tasten zusammen ausgelöst wird, schreibt der Computer immer den amerikanischen Wert für 60 Hz in dieses Bit. Dann geht die Uhr aber empfindlich nach. Man muß also einen Weg finden, der erlaubt, dort in diesem Fall wieder eine 1 einzuschreiben. Das ist durch eine eigene NMI-Routine möglich. Sie sehen schon, der Weg zur Nutzung dieser verlockenden Uhren ist ziemlich dornenreich!

Noch interessanter ist das Bit 7 im CRB. Das Setzen der Uhrzeit ist nämlich nur möglich, wenn dieses Bit den Inhalt 0 hat. Was geschieht, wenn dort eine 1 steht? Dann bestimmt man nicht die aktuelle Uhrzeit, sondern man stellt einen Wecker (das ist die Alarmzeit). Das geschieht nach dem Setzen dieses Bits genauso wie vorhin das Einschreiben der Uhrzeit (also erstaunlicherweise auch in genau dieselben Register!). Im Unterschied dazu ist allerdings ein Lesen der Alarmzeit nicht möglich — das ergibt immer die aktuelle Uhrzeit. Man muß für diesen Fall die Weckzeit irgendwo abspeichern und bei Bedarf dann von dort lesen.

Weil man ja meistens nach dem Erreichen der Alarmzeit irgendeine Reaktion erwartet, ist im ICR (also dem Unterbrechungskontrollregister 13) jedes CIA noch ein Bit reserviert — das Bit 2 —, mit dessen Hilfe der Alarm per IRQ oder NMI wie auch immer geartet losbrechen kann. Der Phantasie sind hier nur wenige Grenzen gesetzt. Wie man mit diesem ICR umgeht, ist Ihnen noch aus der Folge 10 geläufig.

Damit sind wir durch die Eigenheiten der CIAs durch. Man braucht tatsächlich keine Scheu zu haben, diese Echtzeituhren zu nutzen. Lediglich die Uhr im CIA1 wird manchmal verwendet, einen bestimmten Wert für die Zufallszahlenerzeugung zu generieren. Aber das sollte einer eigenen Uhren-Routine nicht in die Quere kommen. Solch eine Echtzeituhr finden Sie im beiliegenden Listing 3 und 4.

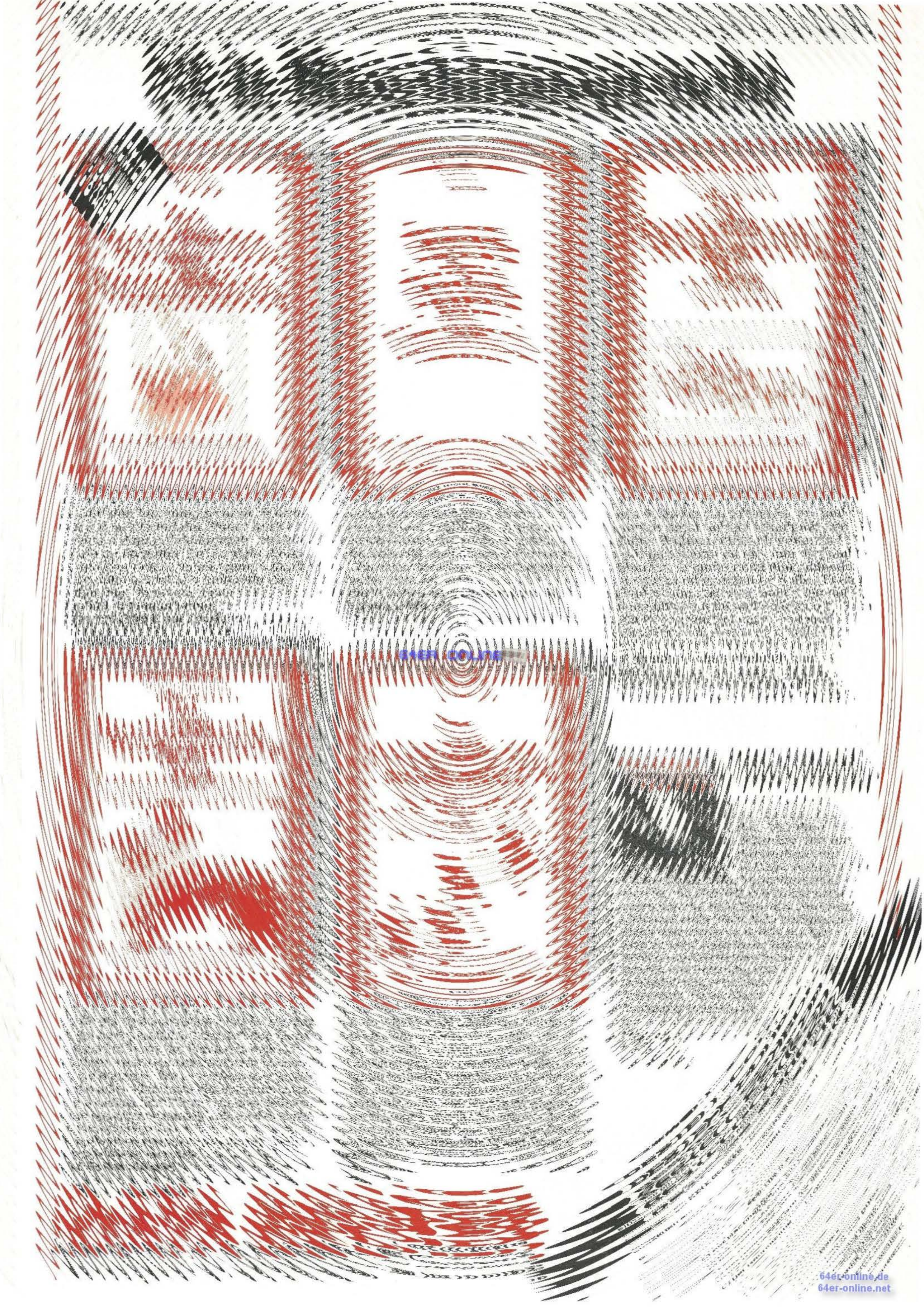
Durch SYS49152 aktivieren Sie die Uhr, die Sie mit SYS49261 auch wieder abschalten können. Durch ein USR-Kommando A=USR (String) stellen Sie die Startzeit ein. String kann dabei eine Stringvariable sein oder auch direkt ein String der Form »HHMMSST« (also Stunden, Minuten, Sekunden, Zehntelsekunden). In A steht eine 0, wenn kein Fehler, aber eine -1, wenn ein Fehler aufgetreten ist. Das Lesen der Uhr erfolgt über ein zweites USR-Kommando: PRINTUSR(Zahl). Dabei kann Zahl eine beliebige Zahl oder Variable sein. Eine Alarmzeit ist ebenfalls einstellbar durch ein USR-Kommando, in dem vor der Zeiteingabe noch ein Buchstabe steht. Beispielsweise stellt A=USR(»A1200000«) einen Wecker auf 12 Uhr. Der Alarm im Programm läßt den Bildschirmrahmen blinken. Abstellen kann man das durch Auslösen eines RESTORE-NMI (also RUN/STOP und RESTORE). Sollten Sie vor dem eingestellten Alarm mal solch einen NMI auslösen, dann muß die Alarmzeit neu gestellt werden. Als Basis für dieses Programm diente ein Listing aus dem schon oft erwähnten Buch von Babel/Krause/Dripke »Das Interface Age Systemhandbuch zum Commodore 64«.

Die Unterbrechungs-Programmierung ist damit abgeschlossen — ebenso diese Serie, die als Einführung in die Assembler-Alchimie nun alle Geheimnisse der Kunst aufgedeckt hat. In den letzten Folgen sind wir schon in die Meistergrade der Zunft aufgestiegen. Vielleicht ging es manchem etwas zu schnell? Dann wird Ihnen der anschließende Kurs »Von Basic zu Assembler« eine Hilfe sein, der behutsam und mit vielen an Basic angelehnten Beispielen die nötige Programmierpraxis vermitteln wird. So wie die Segler sich oft »Mast- und Schotbruch« wünschen, verabschiede ich mich, indem ich Ihnen viele grandiose Abstürze wünsche.

(Heimo Ponnath/gk)

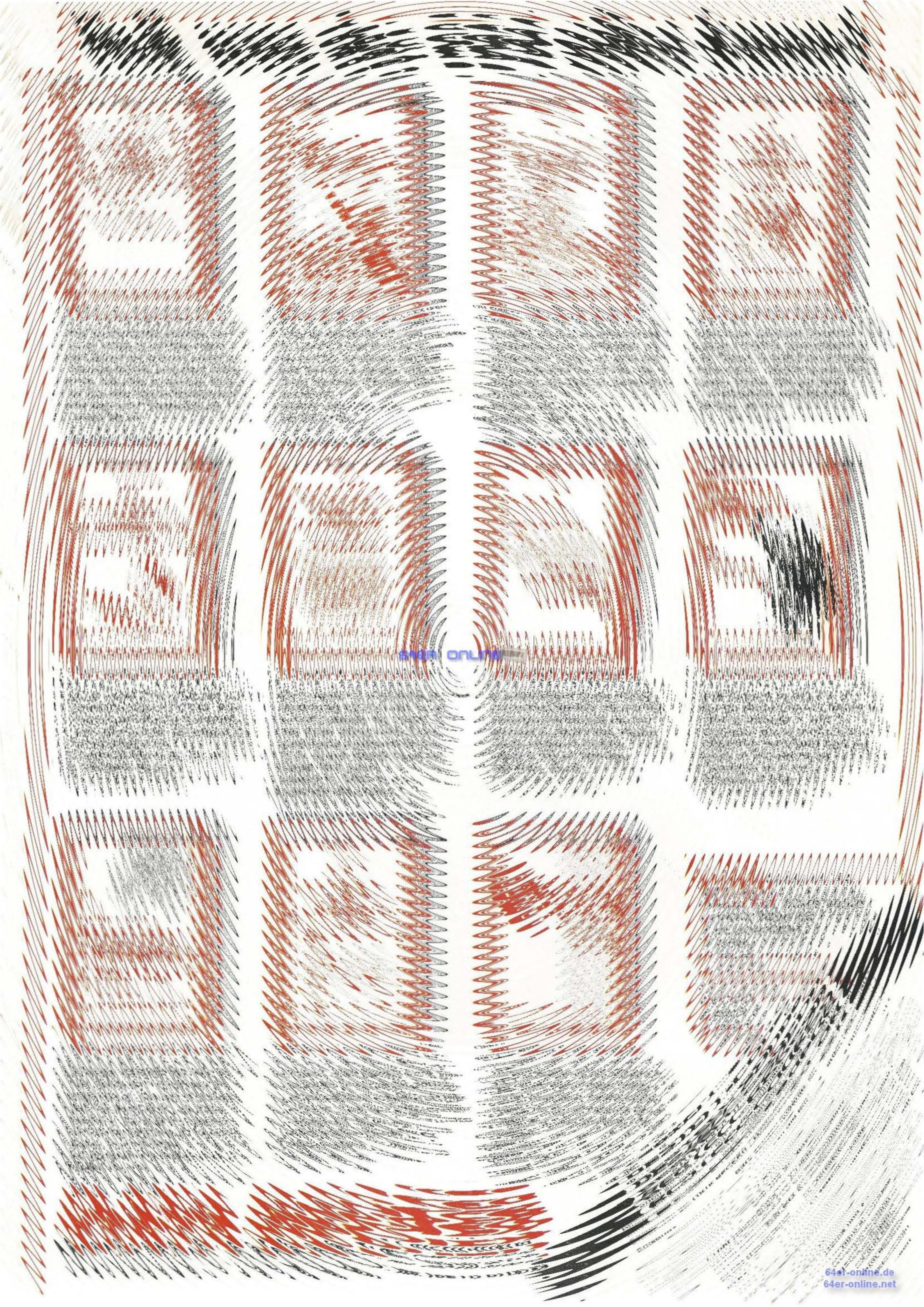
```
PASS 2
7000            0823 ;
7000            084C ;**************************************
7000            0875 ;*                                    *
7000            089E ;*  ECHTZEITUHR MIT ALARMFUNKTION     *
7000            08C7 ;*                                    *
7000            08F0 ;*  LAEUFT MIT DEM NMI-CIA            *
7000            0919 ;*  IN VERBINDUNG MIT DEM IRQ FUER    *
7000            0942 ;*  DEN ALARM                         *
7000            096B ;*                                    *
7000            0994 ;*    HEIMO PONNATH  HAMBURG  1985    *
7000            09BD ;*                                    *
7000            09E6 ;*  (TEILWEISE WURDE EIN PROGRAMM AUS *
7000            0A0F ;*  DEM INTERFACE AGE SYSTEMHANDBUCH  *
7000            0A38 ;*  ZUM COMMODORE 64 , SEITE 114      *
7000            0A61 ;*  ALS BASIS VERWENDET )             *
7000            0A8A ;*                                    *
7000            0AB3 ;**************************************
7000            0AB6 ;
C000            0AC2               .BA $C000
C000            0AC8               .OS
C000            0AC8 ;
C000            0AF4 ;********** ZEROPAGE-LABELS **********
C000            0AF7 ;
C000            0B18 VERZ          .DE $02              ;AKTUELLE VERZOEG.
C000            0B38 FARB          .DE $03              ;WERT FUER RAHMEN
C000            0B57 ;                 EOR-OPERATION
C000            0B78 VORW          .DE $04              ;VERZOEGERUNGSWERT
C000            0B9B VALTYP        .DE $0D              ;INHALT:FF=STR 0=N
C000            0B9E ;
C000            0BAD INDEX         .DE $22
C000            0BC6 INDEX3        .DE $24              ;POINTER
C000            0BD6 INDEX4        .DE $25
C000            0BF5 FAC1          .DE $62              ;1.MANTISSENBYTE
C000            0BF8 ;
C000            0C21 ;********** LABELS PAGES 3 ***********
C000            0C24 ;
C000            0C44 USRADDL       .DE $0311            ;USR-POINTER
C000            0C57 USRADDH       .DE $0312
C000            0C5A ;
C000            0C6A FREI          .DE $0313
C000            0C6D ;
C000            0C8A IRQVL         .DE $0314            ;IRQ-VEKTOR
C000            0C9B IRQVH         .DE $0315
C000            0C9E ;
C000            0CBC NMINVL        .DE $0318            ;NMI-VEKTOR
C000            0CCE NMINVH        .DE $0319
C000            0CD1 ;
C000            0CFA ;********** LABELS INTERPRETER ********
C000            0CFD ;
C000            0D23 ILLQUERR      .DE $B248            ;ILLEGAL QUANTITY
C000            0D4B ;            ERROR=NORMALWERT USR-VEKTOR
C000            0D4E ;
C000            0D70 STRINI8       .DE $B47D            ;SPEICHERPLATZ
C000            0D99 ;            PRUEFEN,STRINGPOINTER SETZEN
C000            0DBF STRLIT67      .DE $B4CA            ;REST DER STRING-
C000            0DD8 ;            LESE-ROUTINE
C000            0DF7 LEN1          .DE $B782            ;STRINGLAENGE
C000            0E11 ;            IN Y-REGISTER
C000            0E32 ACTOFC        .DE $BC3C            ;AKKU NACH FAC
C000            0E35 ;
C000            0E5E ;********** LABELS VIC-II-CHIP ********
C000            0E61 ;
C000            0E7E RAND          .DE $D020            ;RAHMENFARBE
C000            0E81 ;
C000            0EAA ;********** LABELS CIA-BAUSTEINE ******
C000            0EAD ;
C000            0ECA CIA1          .DE $DC00            ;START CIA-1
C000            0EEC ICR1          .DE $DC0D            ;IRQ-KONTROLLREG.
C000            0F0E CRA1          .DE $DC0E            ;TIMER-A KONTRREG
C000            0F11 ;
C000            0F34 TOD10TH2      .DE $DD08            ;1/10 SEKUNDEN
C000            0F51 TODSEC2       .DE $DD09            ;SEKUNDEN
C000            0F6D TODMIN2       .DE $DD0A            ;MINUTEN
C000            0F8F TODHR         .DE $DD0B            ;STUNDEN + AM/PM
C000            0FB1 ICR2          .DE $DD0D            ;NMI-KONTROLLREG.
C000            0FD3 CRA2          .DE $DD0E            ;TIMER-A KONTRREG
C000            0FF5 CRB2          .DE $DD0F            ;TIMER-B KONTRREG
C000            0FF8 ;
C000            0FFB ;
C000            1024 ;********* LABELS OBERES ROM **********
C000            1027 ;
C000            1045 NORM          .DE $EA31            ;NORMALER IRQ
C000            1048 ;
C000            106B TASTFLAG      .DE $F6BC            ;TEIL DER NMI-
C000            108E ;             ROUTINE (KEIN MODUL)
C000            10AD VECTAB        .DE $FD2F            ;TABELLE DER
C000            10C8 ;             ROM-VEKTOREN
C000            10E9 VECTAB7       .DE $FD35            ;MSB DES NMI-
C000            110E ;             VEKTORS IN DER TABELLE
C000            1132 IORESET19     .DE $FDB6            ;I/O-RESET:BEI
C000            1157 ;             SETZEN DES CRA IRQ-CIA
C000            117B NMIXCT16      .DE $FE6C            ;NMI-ROUTINE AB
C000            119D ;             SCREEN-EDITOR-RESET
C000            11C1 NMIRS232      .DE $FE72            ;NMI-ROUTINE AB
C000            11DE ;             RS232-HANDLING
C000            11FF NMIEND        .DE $FEBC            ;ENDE DER NMI-
C000            1215 ;             ROUTINE
C000            1237 STOP          .DE $FFE1            ;KERNAL STOP SPRG
C000            1254 ;             NACH JMP($328)
C000            1257 ;
C000            125A ;
C000            1283 ;********** AKTIVIEREN ****************
C000            1286 ;
C000 A9 8E      12A2 INIT          LDA #L,USR           ;USR-VEKTOR
C002 8D 11 03   12B7               STA USRADDL          ;LADEN
C005 A9 C0      12C4               LDA #H,USR
C007 8D 12 03   12D2               STA USRADDH
C00A            12D5 ;
C00A A9 1D      12F2               LDA #L,NMI           ;NMI-VEKTOR MIT
C00C 8D 18 03   130D               STA NMINVL           ;STARTADRESSE
C00F A9 C0      1327               LDA #H,NMI           ;DER EIGENEN
C011 8D 19 03   1345               STA NMINVH           ;NMI-ROUTINE LAD
C014            1348 ;
C014 AD 0E DD   1365               LDA CRA2             ;BIT7 CRA SETZEN:
C017 09 80      137C               ORA #$80             ;%1000 0000
C019 8D 0E DD   1397               STA CRA2             ;NETZFREQ.=50HZ
C01C            139A ;
C01C 60         13A0               RTS
C01D            13A3 ;
C01D            13CC ;********** EIGENE NMI-ROUTINE ********
C01D            13CF ;
C01D 48         13EF NMI           PHA                  ;ANFANG NORMALE NMI-R.
C01E 8A         1406               TXA                  ;REGISTER RETTEN
C01F 48         140C               PHA
C020 98         1412               TYA
C021 48         1418               PHA
C022            141B ;
C022 A9 7F      1439               LDA #$7F             ;SPERREN ALLER NMI
C024 8D 0D DD   1444               STA ICR2
C027            1447 ;
C027 AC 0D DD   1462               LDY ICR2             ;PRUEFEN OB NMI
C02A 10 06      1481               BPL RESTNMI          ;VOM CIA2 KOMMT.
C02C            14A4 ;                 WENN NEIN=SPRUNG
C02C 4C 6A C1   14C1               JMP ALARM            ;WENN JA, ALARM
C02F 4C 72 FE   14E1 CIANMI        JMP NMIRS232         ;REST DER
C032            1508 ;                 NORMALEN NMI-ROUTINE
C032            150B ;
C032            1534 ;***** EIGENE RESTORE-NMI-ROUTINE *****
C032            1537 ;
C032            155F ;   DIE MODULPRUEFUNG WIRD AUSGELASSEN
C032            1562 ;
C032 20 BC F6   1587 RESTNMI       JSR TASTFLAG         ;TEIL DER NMI-
C035 20 E1 FF   15A5               JSR STOP             ;ROUTINE ZUR STOP-
C038 D0 F5      15C2               BNE CIANMI           ;TASTEN-ABFRAGE
C03A            15C5 ;
C03A A2 04      15E3               LDX #$04             ;IRQ UND BRK VEKT.
C03C BD 2F FD   1606 UMLAD1        LDA VECTAB,X         ;RESTAURIEREN
C03F 9D 13 03   1613               STA FREI,X
C042 CA         1619               DEX
C043 D0 F7      1626               BNE UMLAD1
C045            1629 ;
C045            1651 ;   DER NMI-VEKTOR WIRD UEBERSPRUNGEN
C045            1654 ;
C045 A2 1A      1671               LDX #$1A             ;RESTAURIEREN DER
C047 BD 35 FD   1693 UMLAD2        LDA VECTAB7,X        ;RESTLICHEN
C04A 9D 19 03   16AC               STA NMINVH,X         ;VEKTOREN
C04D CA         16B2               DEX
C04E D0 F7      16BF               BNE UMLAD2
C050            16C2 ;
C050            16E3 ;  ZUNAECHST NORMALER I/O-RESET
C050            16E6 ;
C050 A9 7F      16FD               LDA #$7F             ;= 0111 111
C052 8D 0D DC   171B               STA ICR1             ;SPERREN ALLER IRQ
C055 8D 0D DD   1739               STA ICR2             ;SPERREN ALLER NMI
C058 8D 00 DC   1753               STA CIA1             ;DATENREGISTER
C05B            177A ;                 PORT A AUF NORMALWERT
C05B A9 08      1791               LDA #$08             ;=0000 1000
C05D 8D 0E DC   17AD               STA CRA1             ;TIMER A IM CIA1
C060            17B0 ;
C060            17D2 ;  ERSATZ FUER BELEGUNG DES CRA2
C060            17D5 ;
C060 A9 88      17EC               LDA #$88             ;=1000 1000
C062 8D 0E DD   1809               STA CRA2             ;TIMER A IM CIA2:
C065            1829 ;                 BIT 0 AUF STOP
C065            184F ;                 BIT 3 AUF EINZELLAUF
C065            1875 ;                 BIT 5 SYSTEMTAKT EIN
C065            189D ;                 BIT 7 ECHTZEITUHR=50HZ
C065            18A0 ;
C065            18C0 ;  REST DES NORMALEN I/O-RESET
C065            18C3 ;
C065            18C6 ;
C065 A9 08      18DE               LDA #$08             ;= 0000 1000
C067 20 B6 FD   18EE               JSR IORESET19
C06A            18F1 ;
C06A            1919 ;  REST DER NORMALEN RESTORE-NMI-ROUT.
C06A            191C ;
C06A 4C 6C FE   193A               JMP NMIXCT16         ;EINSPRUNG BEI
C06D            195F ;                 SCREEN EDITOR RESET
C06D            1962 ;
C06D            198B ;**** ABSCHALTEN DER TIME OF DAY UHR **
C06D            198E ;
C06D            19AD ;             DURCH SYS-KOMMANDO
C06D            19B0 ;
C06D A9 48      19D1 AUS           LDA #L,ILLQUERR      ;USR-VEKTOR
C06F 8D 11 03   19EF               STA USRADDL          ;AUF NORMALWERT
C072 A9 B2      1A01               LDA #H,ILLQUERR
C074 8D 12 03   1A0F               STA USRADDH
C077            1A12 ;
C077 78         1A18               SEI
C078 A9 47      1A35               LDA #$47             ;RESTAURIEREN DES
C07A 8D 18 03   1A4F               STA NMINVL           ;NMI-VEKTORS
C07D A9 FE      1A5A               LDA #$FE
C07F 8D 19 03   1A67               STA NMINVH
C082            1A6A ;
C082 A9 31      1A87               LDA #L,NORM          ;RESTAURIEREN
C084 8D 14 03   1AA4               STA IRQVL            ;DES IRQ-VEKTORS
C087 A9 EA      1AB2               LDA #H,NORM
C089 8D 15 03   1ABE               STA IRQVH
C08C            1AC1 ;
C08C 58         1AC7               CLI
C08D 60         1ACD               RTS
C08E            1AD0 ;
C08E            1AF9 ;**** DURCH USR AUFRUFBARE ROUTINE ****
C08E            1AFC ;
C08E 24 0D      1B1D USR           BIT VALTYP           ;WELCHER TYP VON
C090            1B44 ;                 VARIABLEN LIEGT VOR ?
C090 30 03      1B5F               BMI STRING           ;WENN STRING,
C092            1B83 ;                 DANN UEBERSPRINGEN
C092 4C 20 C1   1B9F               JMP ZAHLVAR          ;SONST SPRUNG
C095            1BA2 ;
C095            1BCB ;****** STELLEN DER ECHTZEITUHR *******
C095            1BEB ;             DURCH USR("HHMMSST")
C095            1BEE ;
C095 20 82 B7   1C0F STRING        JSR LEN1             ;Y=STRINGLAENGE
C098 C0 07      1C2C               CPY #$07             ;STRING=7ZEICHEN?
C09A D0 40      1C4A               BNE ALSET            ;NEIN DANN ALARM
C09C            1C69 ;                 ZEIT STELLEN?
C09C            1C6C ;
C09C AD 0F DD   1C88               LDA CRB2             ;TIMER B IN CIA2
C09F 29 7F      1CA3               AND #$7F             ;BIT7 LOESCHEN:
C0A1 8D 0F DD   1CBF               STA CRB2             ;NORMALE UHRZEIT
C0A4            1CE4 ;                 IN ECHTZEITUHR CIA2
C0A4            1CE7 ;
C0A4            1D05 ;  AUSLESEN DES ZEIT-STRINGS
C0A4            1D08 ;
C0A4 A0 00      1D22               LDY #$00             ;ZAEHLER AUF 0
C0A6 A9 24      1D47 STELLEN       LDA #$24             ;BCD 24 STD-VERGL.
C0A8 20 FE C0   1D65               JSR ASCBCD           ;ZEICHENTEST UND
C0AB            1D8D ;                 UMWANDLUNG IN BCD-ZAHL
C0AB D0 02      1DAB               BNE STD12            ;STUNDEN UNGLEICH
C0AD            1DCF ;                 NULL ? DANN SPRUNG
C0AD A9 24      1DE6               LDA #$24             ;SONST = 24
C0AF            1DE9 ;
C0AF C9 13      1E0B STD12         CMP #$13             ;STUNDEN GROESSER
C0B1 90 07      1E2A               BCC STDSET           ;ODER GLEICH 12 ?
```

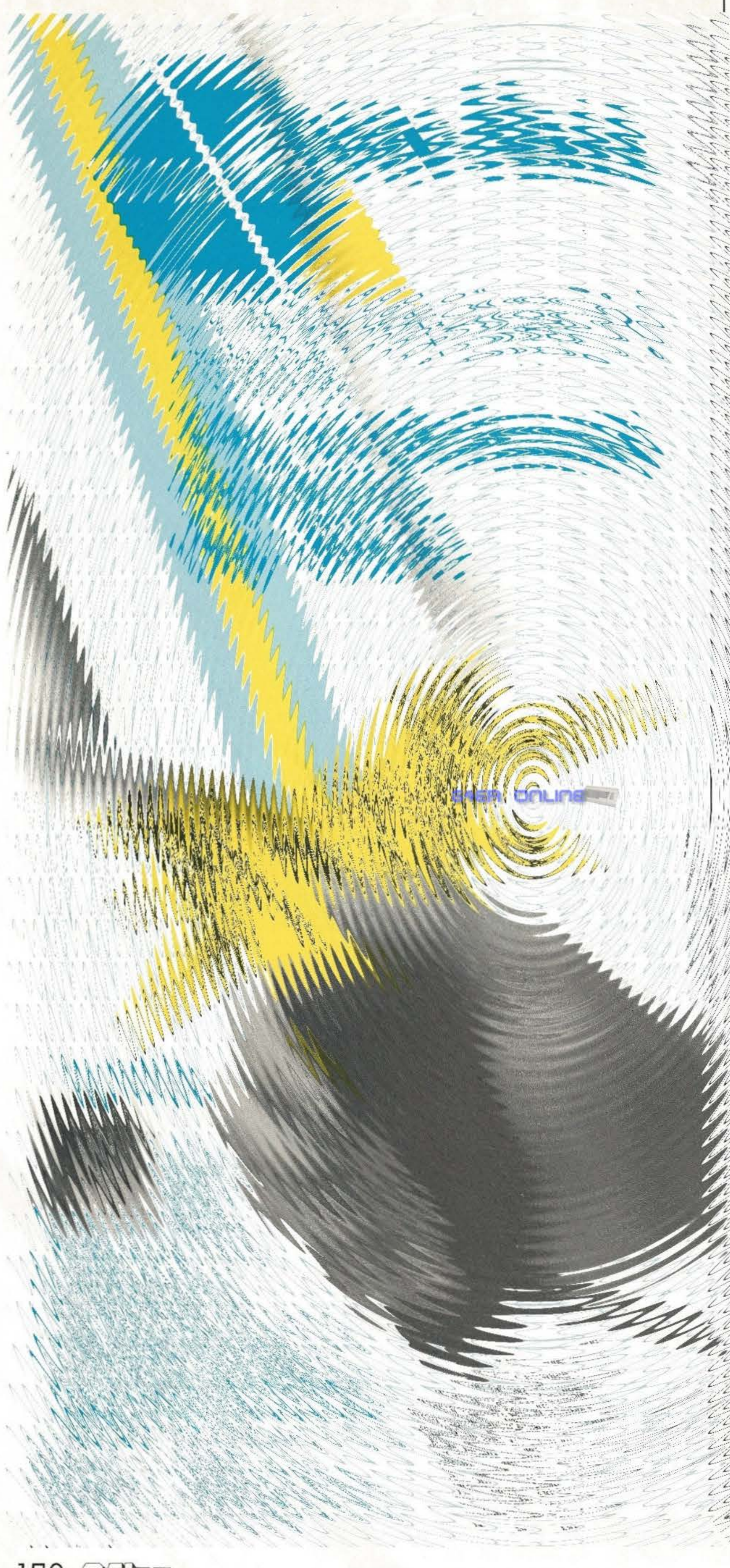
Listing 4. Der Quelltext zur Echtzeituhr



```
C0B3           1E4D ;                NEIN, DANN SPRUNG
C0B3 F8        1E67             SED                  ;SONST DAVON BCD 12
C0B4 38        1E7B             SEC                  ;SUBTRAHIEREN
C0B5 E9 12     1E8B             SBC #$12             ;UND
C0B7 D8        1E91             CLD
C0B8 09 80     1EA9             ORA #$80             ;BIT7 SETZEN
C0BA           1EAC ;
C0BA 8D 0B DD  1ECF STDSET      STA TODHR            ;BCD-STUNDEN UND
C0BD           1EF7 ;                AM/PM-FLAG IN TOD-CIA2
C0BD           1EFA ;
C0BD 20 FC C0  1F18             JSR ASCBCD1          ;ZEICHENTEST U.
C0C0           1F3F ;                UMWANDELN IN BCD-ZAHL
C0C0 8D 0A DD  1F5A             STA TODMIN2          ;ERGEBNIS IN
C0C3           1F7F ;                TOD-MINUTENREGISTER
C0C3           1F82 ;
C0C3 20 FC C0  1F9F             JSR ASCBCD1          ;DASSELBE FUER
C0C6 8D 09 DD  1FBB             STA TODSEC2          ;DIE SEKUNDEN
C0C9           1FBE ;
C0C9 20 66 C1  1FDA             JSR TEST             ;PRUEFEN,OB 1/10
C0CC           1FF9 ;                SEKUNDEN=ZAHL
C0CC 8D 08 DD  2017             STA TOD10TH2         ;UND EINTRAGEN
C0CF           2039 ;                INS TOD-REGISTER
C0CF           205D ;  DIE UHR BEGINNT JETZT ZU LAUFEN
C0CF           2060 ;
C0CF A9 00     207D             LDA #$00             ;KENNUNG FUER OK.
C0D1 4C 3C BC  20A2 AKKUFAC     JMP ACTOFC           ;AKKU ZUR UEBER-
C0D4           20C9 ;                GABE INS BASIC IN FAC
C0D4           20CC ;
C0D4           20F5 ;********** FEHLER AUFGETRETEN ********
C0D4           20F8 ;
C0D4 68        2116 FEHLER      PLA                  ;JSR-ADRESSEN VOM
C0D5 68        212A             PLA                  ;STAPEL HOLEN
C0D6           212D ;
C0D6 68        2138 ERROR       PLA
C0D7 68        213E             PLA
C0D8           2141 ;
C0D8 A9 FF     2164 ERROR1      LDA #$FF             ;FEHLERKENNUNG IN
C0DA D0 F5     2180             BNE AKKUFAC          ;AKKU UND FAC
C0DC           2183 ;
C0DC           21AC ;ENDE DIESES TEILS D. UNBEDINGTEN SPRG.
C0DC           21D5 ;-------------------------------------
C0DC           21D8 ;
C0DC           2201 ;**** ALARMZEIT EINLESEN *************
C0DC           2204 ;
C0DC           222A ;  AUFRUF DURCH Z.B. USR("AHHMMSST")
C0DC           222D ;
C0DC C0 08     224B ALSET       CPY #$08             ; 8 ZEICHEN ?
C0DE D0 F8     2265             BNE ERROR1           ;NEIN=FEHLER
C0E0           2268 ;
C0E0 AD 0F DD  2273             LDA CRB2
C0E3 09 80     228E             ORA #%10000000       ;ALARMBIT
C0E5 8D 0F DD  22A1             STA CRB2             ;SETZEN
C0E8           22A4 ;
C0E8 A9 84     22C0             LDA #%10000100       ;ALARM-NMI
C0EA 8D 0D DD  22D5             STA ICR2             ;ZULASSEN
C0ED           22D8 ;
C0ED A9 3C     22F3             LDA #$3C             ;VERZOEGERUNGS-
C0EF 85 04     230D             STA VORW             ;WERT VORGEBEN
C0F1 85 02     2318             STA VERZ
C0F3 A9 FF     2336             LDA #$FF             ;EOR-WERT VORGEBEN
C0F5 85 03     2341             STA FARB
C0F7 A0 01     235F             LDY #$01             ;BUCHSTABE UEBERL.
C0F9 4C A6 C0  236D             JMP STELLEN
C0FC           2370 ;
C0FC           2373 ;
C0FC           239C ;*************************************
C0FC           23C5 ;UNTERPROGRAMM ZUR UMWANDLUNG DER ASCII
C0FC           23EC ;CODES IN BCD-ZAHLEN UND PRUEFUNG DER
C0FC           23FF ;EINGABE-ZEICHEN.
C0FC           2402 ;
C0FC A9 60     2427 ASCBCD1     LDA #$60             ;BCD 60 ALS GRENZE
C0FE           244F ;                FUER MIN UND SEK WERTE
C0FE           2452 ;
C0FE 85 24     2465 ASCBCD      STA INDEX3
C100 20 13 C1  2482             JSR TEST1            ;PRUEFEN OB ZAHL
C103 0A        249A             ASL                  ;AUS LSB INS MSB
C104 0A        24AB             ASL                  ;SCHIEBEN
C105 0A        24B2             ASL
C106 0A        24B9             ASL
C107 85 25     24D7             STA INDEX4           ;UND ZW.SPEICHER
C109           24DA ;
C109 20 13 C1  24F7             JSR TEST1            ;NAECHSTE ZIFFER
C10C           2510 ;                PRUEFEN
C10C 05 25     252C             ORA INDEX4           ;MSB AUS ZWSP.
C10E           2552 ;                UND LSB ZUSAMMENOREN
C10E C5 24     256F             CMP INDEX3           ;UNTER GRENZW.?
C110 B0 C4     258D             BCS ERROR            ;NEIN=FEHLERAUSG.
C112           2590 ;
C112 60        2596             RTS
C113           2599 ;
C113           25C2 ;** PRUEFUNG OB ASCII-ZAHL VORLIEGT ***
C113           25C5 ;
C113 B1 22     25E8 TEST1       LDA (INDEX),Y        ;ZEICHEN EIN-
C115           2607 ;                LESEN IN AKKU
C115 38        260D             SEC
C116 E9 30     2625             SBC #$30             ;< ASCII 0 ?
C118 90 BA     263D             BCC FEHLER           ;JA=FEHLER
C11A           2640 ;
C11A C9 0A     2659             CMP #$0A             ;>= ASCII : ?
C11C B0 B6     2671             BCS FEHLER           ;JA=FEHLER
C11E           2674 ;
C11E C8        2690             INY                  ;SCHLEIFENZAEHLER + 1
C11F 60        2696             RTS
C120           2699 ;
C120           26C2 ;**** ENDE PROGRAMMTEIL UHR STELLEN ***
C120           26C5 ;
C120           26EE ;-------------------------------------
C120           26F1 ;
C120           271A ;********** UHR LESEN ****************
C120           271D ;
C120           273E ;      GESCHIEHT DURCH USR(ZAHL)
C120           2741 ;
C120 A9 07     2761 ZAHLVAR     LDA #$07             ;STRINGLAENGE
C122 20 7D B4  277F             JSR STRINI8          ;SCHAFFT 7 BYTE
C125           27A7 ;                PLATZ FUER STRING UND
C125           27CF ;                LEGT START NACH $62/63
C125           27F6 ;                SOWIE LAENGE NACH $61
C125           2814 ;                (FAC $61-66)
C125           2817 ;
C125 A0 00     2831             LDY #$00             ;ZAEHLER AUF 0
C127 AD 0B DD  284F             LDA TODHR            ;STUNDE AUSLESEN,
C12A           2878 ;                DABEI WIRD GESAMTE ZEIT
C12A           28A1 ;                ZWISCHENGESPEICHERT UND
C12A           28C7 ;                ERST NACH LESEN DER
C12A           28EF ;                1/10-SEK ZURUECKGEHOLT
C12A           2914 ;                MIT AKTUELLEN WERT.
C12A           2917 ;
C12A 08        2935             PHP                  ;STATUS ZWISCHENSPEICH.
C12B           2938 ;
C12B 29 1F     294F             AND #$1F             ;=0001 1111
C12D           2978 ;                LOESCHEN DER AM/PM-FLAG
C12D C9 12     2996             CMP #$12             ;=0001 0010 =BCD12
C12F D0 02     29B0             BNE NO12             ;<>DANN SPRUNG
C131 A9 00     29CE             LDA #$00             ;SONST STATTDESSEN
C133           29D1 ;
C133 28        29F0 NO12        PLP                  ;STATUS ZURUECKHOLEN
C134 10 05     2A0A             BPL AM               ;FALLS KEINE AM/
C136           2A2F ;                PM-FLAG GESETZT WAR
C136           2A32 ;
C136 F8        2A4C             SED                  ;SONST ADDIEREN VON
C137 18        2A62             CLC                  ;BCD 12 WEIL PM
C138 69 12     2A6D             ADC #$12
C13A D8        2A73             CLD
C13B           2A76 ;
C13B 20 55 C1  2A96 AM          JSR BCDASC           ;UP ZUR UMRECHNG
C13E           2ABC ;                VOM BCD IN ASCII UND
C13E           2AE5 ;                ABLEGEN IM STRING. HIER
C13E           2B02 ;                STUNDENWERT
C13E           2B05 ;
C13E AD 0A DD  2B22             LDA TODMIN2          ;DASSELBE FUER
C141 20 55 C1  2B3C             JSR BCDASC           ;MINUTENWERT
C144           2B3F ;
C144 AD 09 DD  2B5C             LDA TODSEC2          ;UND SEKUNDEN-
C147 20 55 C1  2B6F             JSR BCDASC           ;WERT
C14A           2B72 ;
C14A AD 08 DD  2B8C             LDA TOD10TH2         ;UND 1/10-
C14D 20 60 C1  2BA8             JSR BCDASC1          ;SEKUNDENWERT
C150           2BAB ;
C150 68        2BC7             PLA                  ;USR-STRING-ARGUMENT-
C151 68        2BE5             PLA                  ;RUECKSPRUNG VORBEREIT.
C152           2BE8 ;
C152 4C CA B4  2C06             JMP STRLIT67         ;BRINGT STRING
C155           2C29 ;                DESCRIPTOR IN DIE
C155           2C52 ;                DESCRIPTORTABELLE ($19-
C155           2C79 ;                $21),SETZT POINTER IN
C155           2CA2 ;                FAC(HIER $64/65) DARAUF
C155           2CC9 ;                SETZT STRING-FLAGGE,
C155           2CE9 ;                ERHOEHT LETZTEN
C155           2D10 ;                DESCRIPTOR-INDEX UM 3
C155           2D38 ;                ROUTINE ENDET MIT RTS.
C155           2D3B ;
C155           2D64 ;**** ENDE DES LESENS DER UHR ********
C155           2D67 ;
C155           2D90 ;-------------------------------------
C155           2D93 ;
C155           2DBC ;* UNTERPROGRAMM Z. UMRECHNUNG BCD IN *
C155           2DE5 ;* ASCII U.EINTRAGEN IN STRINGSPEICHER*
C155           2DE8 ;
C155 48        2E0B BCDASC      PHA                  ;AUF STAPEL ZW.SPEICH.
C156           2E0E ;
C156 4A        2E2A             LSR                  ;MSB INS LSB SCHIEBEN
C157 4A        2E31             LSR
C158 4A        2E38             LSR
C159 4A        2E3F             LSR
C15A           2E42 ;
C15A 20 60 C1  2E5E             JSR BCDASC1          ;IN ASCII UM-
C15D           2E85 ;                RECHNEN UND SPEICHERN
C15D           2E88 ;
C15D 68        2EA5             PLA                  ;ZURUECKHOLEN DER BCD-
C15E 29 0F     2EC3             AND #$0F             ;ZAHL,LOESCHEN DES
C160           2ED8 ;                MSB
C160           2EDB ;
C160 09 30     2F00 BCDASC1     ORA #$30             ;DAZUODERN VON $30
C162           2F28 ;                ERZEUGT (WEIL NUR ZAHL
C162           2F4F ;                ZWISCHEN 0 UND 9) DEN
C162           2F78 ;                ASCII-WERT ($30 BIS 39)
C162           2F7B ;
C162 91 62     2F98             STA (FAC1),Y         ;EINTRAGEN IN
C164           2FB7 ;                STRINGTABELLE
C164           2FBA ;
C164 C8        2FCC             INY                  ;ZAEHLER +1
C165           2FCF ;
C165 60        2FD5             RTS
C166           2FD8 ;
C166           3001 ;****** REST DES UP ASCII-BCD ********
C166           3004 ;
C166 20 13 C1  3020 TEST        JSR TEST1            ;PRUEFT AUF
C169           3042 ;                ASCII-ZAHL (0-9)
C169 60        3048             RTS
C16A           304B ;
C16A           3074 ;****** NMI-REAKTION AUF ALARM ********
C16A           3077 ;
C16A A9 77     3097 ALARM       LDA #L,ALIRQ         ;NEUER IRQ-
C16C 8D 14 03  30AB             STA IRQVL            ;VEKTOR
C16F A9 C1     30BA             LDA #H,ALIRQ
C171 8D 15 03  30C6             STA IRQVH
C174           30C9 ;
C174 4C BC FE  30E5             JMP NMIEND           ;REST DER NOR-
C177           3108 ;                MALEN NMI-ROUTINE
C177           310B ;
C177           3134 ;****** DIE  NEUE IRQ-ROUTINE *********
C177           3151 ;           RAHMENBLINKEN
C177           3154 ;
C177           3157 ;
C177 C6 02     3175 ALIRQ       DEC VERZ             ;ZEITSCHLEIFE
C179 F0 03     3191             BEQ BLINK            ;BLINKEN WENN 0
C17B           3194 ;
C17B 4C 31 EA  31B2             JMP NORM             ;SONST NORMALE IRQ
C17E           31B5 ;
C17E A5 04     31D5 BLINK       LDA VORW             ;ZAEHLER RUECK-
C180 85 02     31E8             STA VERZ             ;SETZEN
C182           31EB ;
C182 AD 20 D0  3203             LDA RAND             ;RAHMENFARBE
C185 45 03     321B             EOR FARB             ;INVERTIEREN
C187 8D 20 D0  3226             STA RAND
C18A           3229 ;
C18A 4C 31 EA  3244             JMP NORM             ;ZUM NORMAL-IRQ
C18D           3247 ;
C18D           324D             .EN
```

Listing 4. Der Quelltext zur Echtzeituhr (Schluß)





Hallo Hardware-Entwickler!

Haben Sie eine nützliche Erweiterung für den C 64 oder C 128 selbst entwickelt? Ob EPROM-Brenner, Stereo-Interface, Video-Signalverstärker, Floppy-Speeder oder Centronics-Interface — senden Sie uns Ihre Bauanleitung. Die 64'er ist auch in dieser Beziehung ein Magazin für Computer-Fans, bei der aktive Mitarbeit großgeschrieben wird! Ihre Entwicklung in der 64'er — reizt Sie der Gedanke nicht?

Wie bei den Listings benötigen wir auch für jede Hardware-Entwicklung ein paar Unterlagen:

- Ausführliche Beschreibung der Entwicklung
- Ein fertig aufgebautes Probeexemplar
- Technische Beschreibung, aus der das Funktionsprinzip der Schaltung hervorgeht
- Auf- und Einbauanleitung
- Ein reprofähiges Platinen-Layout, möglichst im Maßstab 1:1 und 2:1
- Illustrieren Sie die Beschreibungen, wo nötig, durch Zeichnungen und Grafiken
- Alle Entwicklungen müssen von Ihnen selbst stammen, die Verwendung von Schaltungen (auch teilweise) aus Büchern und anderen Publikationen ist nicht möglich. Auch wenn Sie selbst nichts entwickelt haben, sind wir an Ihrer Meinung interessiert. Schreiben Sie uns, welche Erweiterung Ihrer Meinung nach am wichtigsten ist.

Wenn Sie sich an der Gestaltung unseres Magazins mit Hardware-Bauanleitungen beteiligen wollen, schreiben Sie einfach an:

Markt & Technik
Verlag Aktiengesellschaft
Redaktion 64'er
Herrn Arnd Wängler
Hans-Pinsel-Str. 2
8013 Haar bei München

Comal 80 — die universelle Programmiersprache

Hand aufs Herz: Wie oft haben Sie sich als C 64-Programmierer schon über die Unzulänglichkeiten des eingebauten Basic 2.0 geärgert und sich ein komfortableres Basic gewünscht? Diese Zeiten sind jetzt endgültig vorbei. Mit Comal 80 nutzen Sie die enormen Hardware-Fähigkeiten des C 64 voll aus, und zwar ohne Maschinensprache, ohne PEEKs und POKEs, dafür schneller, komfortabler und übersichtlicher, als Sie es sich je vorgestellt haben. Comal vereinigt die leichte Erlernbarkeit von Basic mit der übersichtlichen Strukturierung von Pascal und den grafischen Fähigkeiten von Logo. Das Ergebnis ist eine neue Sprache, die alles, was es bisher an Programmiersprachen für den C 64 gibt, in den Schatten stellt.

Vergessen Sie alles, was Sie über die Programmierung Ihres C 64 wissen! Denn mit Comal 80 verwandelt sich der C 64 in einen neuen Computer.

64 KByte im ROM-Modul

Comal 80 ist für den C 64 nur als Steckmodul erhältlich. Dieses Modul enthält 64 KByte ROM, die in vier Speicherbänken zu je 16 KByte organisiert sind. Die eingebaute Bank-Auswahl-Logik schaltet ganz nach Bedarf zwischen den vier Comal-ROM-Bänken, dem Basic-ROM und dem darunterliegenden RAM hin und her. Bild 1 zeigt die Speicheraufteilung des C 64 unter Comal 80. Der gesamte adressierbare Speicherbereich (RAM und ROM) beträgt 152 KByte (!), für Comal-Programme stehen davon 30 KByte zur Verfügung. Das entspricht dem auch bei Basic-Erweiterungen wie Simons Basic oder Exbasic Level II vorhandenen Programm-Speicherplatz. Dazu kommen nochmals 16 KByte RAM, die für selbstgeschriebene Spracherweiterung (User Packages) verfügbar sind. Wem dieser Speicherplatz noch nicht ausreicht, kann sich Comal-Erweiterungen mit einem Umfang von bis zu 32 KByte entweder in Assembler oder in Comal selbst schreiben und in ein EPROM brennen — das Comal-Modul enthält zu diesem Zweck einen freien EPROM-Steckplatz.

Der bildschirmorientierte Editor des C 64 wurde in vielen Punkten noch weiter verbessert. Zahlreiche Control-Funktionen wie »lösche ab Cursor-Position bis Zeilenende« oder »Cursor auf vorhergehendes Wort setzen« erleichtern die Programmeingabe. Spezielle Comal-Kommandos wie DEL (Zeilenbereich löschen), AUTO (automatische Zeilennummern-Vorgabe) oder RENUM (Neunumerierung eines Comal-Programms) bringen großen Komfort bei der Programm-Eingabe und -Änderung. Die Funktionstasten sind mit sinnvollen Kommandos belegt, können aber vom Anwender beliebig umdefiniert werden.

Komfortabler Editor

Comal ist genaugenommen weder ein Interpreter noch ein Compiler, sondern eine Zwischenstufe. Comal-Programme werden vor der Ausführung (die interpretativ erfolgt) in zwei Stufen »vor-übersetzt«. Die erste Stufe ist die Umsetzung der Comal-Schlüsselworte in sogenannte »Token«, also meist aus einem Byte bestehende Befehls-Abkürzungen. Eine solche Vorübersetzung findet auch beim normalen Basic 2.0 statt. Comal geht aber noch weiter: Wird ein Comal-Programm gestartet, dann führt Comal vor dem eigentlichen Programmlauf eine semantische Analyse durch, in deren Verlauf die effektiven Adressen von Variablen, Prozeduren und Funktionen ermittelt und gespeichert werden. Während des Programmlaufs braucht Comal daher nicht lange in Variablentabellen zu suchen, sondern findet die gesuchte Adresse praktisch unmittelbar vor. Dadurch werden Ausführungsgeschwindigkeiten erreicht, die in der Regel erheblich über denen vergleichbarer Basic-Programme liegen. Einen Compiler kann man bei Comal getrost vergessen, er würde keinen nennenswerten Geschwindigkeitsvorteil mehr bringen.

Von Basic zu Comal

In Comal 80 sind die besten Eigenschaften der drei Programmiersprachen Basic, Pascal und Logo zu einem sehr komfortablen Programmier-System zusammengefaßt. Auf der Kommando-Ebene braucht der von Basic kommende Comal-Anwender nur wenig umzulernen: Programme werden mit LOAD geladen, mit SAVE gespeichert und mit LIST gelistet. Auch Befehle wie NEW, RUN, AUTO, RENUM oder MERGE dürften dem Basic-Programmierer nicht unbekannt sein.

Viele im Programm-Modus verwendete Befehle sind an Basic angelehnt. Das geht so weit, daß Comal vielfach Basic-Befehle akzeptiert, wo eigentlich spezielle Comal-Schlüsselwörter stehen müßten. Ein Beispiel ist die FOR-Schleife, die in Comal statt mit NEXT mit ENDFOR abgeschlossen werden muß. Tippt man jedoch versehentlich »NEXT«, so macht das nichts aus: Das Comal-System ist intelligent genug, um zu erkennen, was der Benutzer mit NEXT meinte und ersetzt das NEXT stillschweigend durch ENDFOR.

Strukturiert wie Pascal

Comal 80 für den C 64 ist in vieler Hinsicht mit Pascal vergleichbar, ohne allerdings dessen in vielen Fällen unnatürliche Strenge und Kompliziertheit zu übernehmen. Comal versucht stets, dem Benutzer dort Arbeit abzunehmen, wo dies von einem intelligenten System auch getan werden kann. Beispielsweise zwingt Pascal dazu, alle Variablen, Konstanten oder Prozeduren vor der ersten Verwendung des entsprechenden Namens zu definieren. Bei Comal dagegen ist es gleichgültig, wo eine Prozedur im Programm definiert wird. Bevor das RUN-Kommando ausgeführt wird, ordnet Co-

mal jedem Auftreten eines Prozedur-Namens die entsprechende Prozedur selbst zu, egal, ob diese vorne, hinten oder in der Mitte eines Programms steht.

Comal 80 ist eine block-strukturierte Sprache und sorgt dadurch für Lesbarkeit der Programme. Die Sprachstruktur wird durch die Form der Bildschirmausgaben beim Listen noch unterstützt. Comal rückt verschachtelte Strukturen automatisch ein, ohne dafür zusätzlichen Speicherplatz zu brauchen. So werden schon optisch von vornherein klare Gliederungen und Strukturen geschaffen, die sich durch Kommentare und Leerzeilen noch vertiefen lassen.

Der Sprachumfang von Comal 80 umfaßt praktisch alle Elemente der strukturierten Programmierung:

Die von Basic her bekannte, vergleichsweise primitive IF-THEN-Abfrage ist in Comal dahingehend erweitert worden, daß ganze Programmteile bedingt ausgeführt werden können. Hierzu wird die IF-Anweisung in der Form IF-THEN-ENDIF verwendet, wobei zwischen THEN und ENDIF ein beliebig umfangreicher Programmteil stehen kann, der nur ausgeführt wird, wenn die Bedingung nach IF wahr ist. Im Falle zweier alternativer Möglichkeiten kann die IF-Anweisung durch ELSE zur Form IF-THEN-ELSE-ENDIF ergänzt werden. Weitere Alternativen lassen sich durch zusätzliche Abfragen mittels ELIF (eine Wortbildung aus ELSE IF) leicht abfragen. Der folgende Programmteil druckt einen Wetterbericht in Abhängigkeit vom Inhalt der Variablen »WETTER«:

```
100 IF WETTER = -1
110     PRINT "schlechtes Wetter"
120   ELIF WETTER = 1
130     PRINT "gutes Wetter"
140   ELSE
150     PRINT "veraenderlich"
160 ENDIF
```

Falls mehrere Bedingungen zu prüfen sind, ist die CASE-Konstruktion allerdings meist übersichtlicher. Der gleiche Wetterbericht mittels CASE-Anweisung formuliert, sieht so aus:

```
100 CASE WETTER
110   WHEN -1
120     PRINT "schlechtes Wetter"
130   WHEN 1
140     PRINT "gutes Wetter"
150   OTHERWISE
160     PRINT "veraenderlich"
170 ENDCASE
```

Für die Programmierung von Schleifen stehen neben der bereits angesprochenen FOR-ENDFOR-Schleife (entspricht FOR-NEXT in Basic) drei weitere Schleifen-Konstruktionen zur Verfügung:

LOOP-ENDLOOP ist eine Endlosschleife, die nur durch einen eingefügten EXIT-Befehl verlassen werden kann.

REPEAT-UNTIL wird durchlaufen, bis die Bedingung hinter UNTIL erfüllt ist.

WHILE-ENDWHILE dagegen prüft zuerst die hinter WHILE stehende Bedingung. Der Programmteil zwischen WHILE und ENDWHILE wird solange ausgeführt, wie die Bedingung wahr ist.

Für ganz spezielle Zwecke existiert auch noch ein Sprungbefehl (GOTO), jedoch nur auf ein mit einem Namen versehenes Sprungziel (LABEL).

Ebenfalls von Pascal übernommen wurde das Konzept der Prozeduren und Funktionen. Auch hier fällt wieder angenehm auf, daß auf einen puren Formalismus zugunsten erhöhter Anwenderfreundlichkeit verzichtet wurde.

Prozeduren und Funktionen

Eine Prozedur ist entfernt mit einem Unterprogramm in Basic verwandt, nur wird sie nicht über »GOSUB (Zeilennummer)«, sondern direkt mit einem Namen aufgerufen. Im Gegensatz zu Basic können dabei alle benötigten Parameter mit übergeben werden. Dies können Zahlen, Strings oder sogar ganze Felder sein. Die Prozedur legt sich dann lokale Variablen an, was bedeutet: Die Namen der Variablen sind nur innerhalb der Prozedur gültig, der gleiche Variablenname bezeichnet also in Prozedur und Hauptprogramm verschiedene Variable. Im übrigen wird, wie in Pascal, zwischen reinen Werteparametern und sogenannten Variablenparametern unterschieden.

Durch Prozeduren bietet Comal die Möglichkeit, den Sprachumfang praktisch beliebig zu erweitern. Ähnlich verhält es sich mit den Funktionen, die — anders als Basic-Funktionen — über beliebig viele Zeilen definiert werden können. Die Funktionsdefinition wird eingeleitet durch das Schlüsselwort FUNC und beendet durch ENDFUNC. Wie bei Prozeduren können beliebig viele Parameter an eine Funktion übergeben werden. Der Unterschied zu Prozeduren besteht darin, daß eine Comal-Funktion immer einen bestimmten Wert zurückliefert, der durch RETURN (Wert) ausgewählt wird. RETURN hat hier nichts zu tun mit dem Basic-Befehl RETURN, sondern definiert nur den von der Funktion zurückzuliefernden Wert.

Fehlerbehandlung leicht gemacht

Um eine möglichst weitgehende Benutzerfreundlichkeit zu realisieren und die Programmierung zu vereinfachen, wurde Comal mit komfortablen Möglichkeiten zur Behandlung von Fehlern versehen. Diese beschränken sich nicht nur auf das Abfangen aller möglichen Fehlerarten, sondern es sind auch automatische Korrekturen und die Weitergabe von Meldungen an übergeordnete Programm-Strukturen möglich. Dabei können sogar bewußt Fehler erzeugt oder übertragen werden. Die Fehlerbehandlung stellt selbst eine Programmstruktur dar. Der Teil des Programms, in dem Fehler abgefangen werden sollen, wird durch TRAP und ENDTRAP eingeklammert. Dieser Programmteil enthält auch die Fehlerbehandlungs-Routine (HANDLER), die im Fehlerfall aufgerufen wird. Ein Beispiel:

```
10 // Quadratwurzel
15 PRINT "bitte Zahl eingeben:"
20 TRAP
```

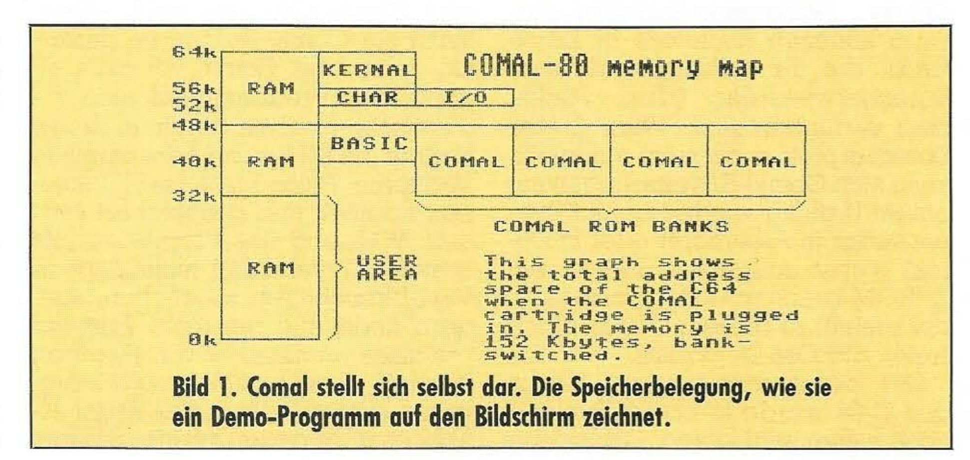


Bild 1. Comal stellt sich selbst dar. Die Speicherbelegung, wie sie ein Demo-Programm auf den Bildschirm zeichnet.



```
25   INPUT ZAHL
30   PRINT SQR(ZAHL)
35  HANDLER
40   CASE ERR OF
45     WHEN 206
50       PRINT "nur Zahl einge-
ben !"
55     WHEN 1
60       PRINT "negati-
ve Zahl nicht erlaubt !"
65     OTHERWISE
70       PRINT ERR, ERRTEXT$
75   ENDCASE
80 ENDTRAP
```

Dieses Programm wartet auf die Eingabe einer Zahl und berechnet danach die Quadratwurzel aus dieser Zahl. Zwei Fehlermöglichkeiten sind dabei offensichtlich: Zum einen kann der Benutzer irrtümlich irgendwelche Buchstaben mit eingeben (Fehler 206, »numeric constant expected«), zum anderen darf das Argument der Quadratwurzel-Funktion nicht negativ werden (Fehler 1, »Argument Error«). Durch den TRAP-HANDLER werden diese beiden Fehler abgefangen und mit entsprechenden Meldungen quittiert. Falls wider Erwarten irgendein anderer Fehler auftreten sollte, so wird in Zeile 70 einfach die Fehlernummer samt dazugehörigem Fehlertext übergeben. Wenn Sie sich an den englischen Fehlermeldungen stören sollten — Comal ermöglicht es Ihnen, alle Fehlermeldungen neu zu formulieren und abzuspeichern.

Comal und die 1541

Ein Schwachpunkt im C 64-System ist bekanntlich die Floppy-Station 1541. Die meisten C 64-Besitzer haben sich wohl zähneknirschend daran gewöhnt, daß das Auflisten des Directorys mit Programmverlust bestraft wird und daß man zur Anzeige der DOS-Fehlermeldungen ein Programm benötigt. Mit Comal 80 hat es ein Ende mit dem Zähneknirschen: DIR oder CAT zeigen das Directory, PASS sendet Kommandos an die 1541. Der Fehlerkanal wird entweder sofort angezeigt — wobei dies mit Hilfe des TRAP-Handlers gesteuert werden kann — oder kann durch STATUS abgefragt werden. CREATE legt eine relative Datei an, mit GET$ kann eine beliebige Anzahl von Bytes von der Floppy-Station eingelesen werden.

Auch in Richtung Programmiertechnik hat man sich einiges einfallen lassen. Programmteile können mit CHAIN nachgeladen werden, wodurch Comal-Programme theoretisch bis zu 200 KByte lang sein können (30 KByte im RAM, 170 KByte auf Diskette). Außerdem können Comal-Prozeduren und -Funktionen als EXTERNAL deklariert werden. Dies bedeutet, daß im Programm nur der Prozedurkopf vorhanden sein muß. Die Prozedur selbst befindet sich in diesem Fall auf der Diskette und wird beim Aufruf automatisch nachgeladen und ausgeführt. Nach getaner Arbeit wird die Prozedur dann einfach wieder aus dem Speicher geworfen, um keinen Platz zu verschwenden.

Comal kann mit bis zu acht Laufwerken gleichzeitig arbeiten und unterstützt von vorneherein auch den Anschluß der »großen« Commodore-Laufwerke der 40xx- und 80xx-Serie.

Die Zusatz-Pakete

Natürlich läßt sich auch mit der Datasette arbeiten, allerdings mit den von Basic bekannten Einschränkungen hinsichtlich Bedienungskomfort und Leistungsfähigkeit.

Bisher war nur von dem international genormten Grundwortschatz von Comal die Rede. Es ist jedoch auch möglich, systemabhängige oder anwenderspezifische Spracherweiterungen zu integrieren. Eine solche Spracherweiterung wird in Comal als »Package« bezeichnet. Packages (Pakete) sind nichts anderes als Sammlungen von Comal-Prozeduren oder -Funktionen, die meist in Maschinensprache geschrieben sind, aber natürlich auch in Comal verfaßt sein können. Eine Reihe derartiger Spezial-Pakete sind in Comal 80 bereits fest eingebaut. Es bereitet aber nur wenig Mühe, sich weitere Packages selbst zusammenzustellen.

Ein fertig programmiertes Package kann auf Diskette abgelegt und bei Bedarf in einen reservierten 16-KByte-RAM-Bereich geladen werden, belastet also nicht den 30 KByte großen Comal-Programmspeicher. Fest eingebaut sind bereits elf Packages, mit denen sich die Fähigkeiten des Comal-Systems nochmals potenzieren:

1. English
Nach »USE ENGLISH« gibt Comal alle Fehlermeldungen im englischen Klartext aus.

2. Danish
Hiermit erscheinen alle Fehlermeldungen im dänischen Original.

3. Graphics
Dieses Package erweitert Comal um eine Reihe von speziell auf den C 64 zugeschnittenen Grafik-Befehlen.

4. Turtle Graphics
An diesem Package werden Logo-Freunde ihre helle Freude haben. Alle Grafikbefehle von Logo stehen hiermit zur Verfügung.

5. Sprites
Alles, was der C 64 in dieser Richtung hardwaremäßig zu bieten hat, ist hier in Form komfortabler Comal-Befehle verfügbar.

6. Sound
Es gibt wohl keine Funktion des SID-Chips, die nicht durch spezielle Comal-Befehle unterstützt wird. Als besonderer Komfort können Noten als Strings codiert und abgespielt werden. Dies geschieht interruptgesteuert, so daß Hintergrundmusik sehr einfach programmiert werden kann.

7. Paddles
Dieses Package enthält Prozeduren und Funktionen zur Abfrage der Controlports.

8. Joysticks
Hiermit wird die Joystick-Kontrolle zum Kinderspiel.

9. Lightpen
Wer einen Lightpen sein eigen nennt, der findet hier alle Spezial-

Fortsetzung auf Seite 164



Bild 2. Ein Beispiel für die grafischen Fähigkeiten von Comal. Verschiedene Windows mit Text und Grafik wurden definiert.





Aufgeräumt mit Mainfile II

Bringen Sie Ordnung in Ihre Karteikästen und Adressenbücher! Mainfile II ist ein Hilfsmittel zur einfachen Programmierung von Dateiverwaltungen. Das Besondere: es belegt keinerlei Speicher im Computer.

Während herkömmliche Programme zur Dateierstellung und Dateiverwaltung sequentielle oder relative Dateien anlegen, folgt Mainfile II wegen der mit den genannten Dateitypen verbundenen Nachteile einem völlig anderen Prinzip: Mainfile II ist eine von Grund auf neue Betriebssystemerweiterung für Commodore-Diskettenlaufwerke. Mainfile II ist völlig unabhängig vom Commodore-Computer. Voraussetzung an den Computer ist lediglich, daß die Floppy 1541 anschließbar ist, da sich alles in deren RAM abspielt. Einsetzbar ist Mainfile II bisher für C 16, VC 20, C 64, SX 64, C 116 und Plus/4!

Mit Mainfile II können pro Diskette bis zu 114 Dateien erstellt werden. Eine Datei besteht dabei aus einzelnen Datensätzen, auf die man extrem einfach und schnell zugreifen kann. Eine Diskette kann maximal 650 Datensätze zu je 228 Byte enthalten.

Bevor man mit Mainfile II zu arbeiten beginnt, muß man sich eine Datendiskette mit den vier sogenannten Makrobefehlen erstellen. Alle Makrobefehle sind durch ein vorangestelltes &-Zeichen kenntlich gemacht. Dem Kopieren dieser Befehle dient ein auf der Systemdiskette befindliches Programm.

Ungewohnt bei Mainfile II ist, daß keinerlei Programme in den Computer geladen werden müssen. Zuerst einmal ist ein Öffnen der Kanäle 2 und 15 zur Übertragung von Daten, Befehlen und Meldungen an die Floppy erforderlich. Mit PRINT #15,"&CREATE,Test" wird der Makrobefehl CREATE aufgerufen, der eine neue Datei mit dem Namen »Test« auf der Diskette einrichtet.

Die Befehlsfolge PRINT#15,"& MAINFILE,Test" aktiviert das Mainfile II-System und gibt den Namen der Datei an, auf die zugegriffen werden soll.

Der einfachste Weg, sich in Mainfile II einzuarbeiten, ist wohl das Laden des Demo-Programmes von der Systemdiskette. Nachdem das in Basic geschriebene Demonstrationsprogramm zur Erstellung und Bearbeitung einer Adreßdatei gestartet wurde, wird man aufgefordert, die Datendiskette einzulegen.

Nur beschränkt durch die Kapazität der Diskette wäre es theoretisch möglich, 650 Datensätze mit Name, Vorname, Straße, Wohnort und Telefonnummer einzutragen. Namen dürfen leider nicht mehrfach vorkommen, da sie nicht unterschieden werden können. Jeder Datensatz wird unmittelbar nach der Eingabe über den Kanal #2 zur Floppy übertragen und in der Datei, deren Name vorher mit »Demo« definiert wurde, gespeichert.

Die eingegebenen Datensätze können anschließend bearbeitet oder nach bis zu acht Kriterien durchsucht werden. Zum Auswerten der Datensätze definiert man Suchbegriffe, indem man zum Beispiel »1 München« eingibt. Anschließend kann man mit dem Kriterium »1« alle Adressen, in denen das Wort »München« vorkommt, suchen und zur Bearbeitung ausgeben lassen. Nachdem man so Mainfile II etwas erprobt hat, sollte man sich das Listing des Demo-Programmes vornehmen. Es ist so klar und übersichtlich geschrieben, daß man die praktische Verwendung aller wesentlichen Befehle sehr schnell versteht (gewisse Grundkenntnisse der Programmiersprache Basic vorausgesetzt). Die klare Gliederung ergibt sich insbesondere durch die hier konsequent benutzte Unterprogrammtechnik.

Mainfile II benutzt die binäre Datenspeicherung: Der zuerst eingegebene Datensatz wird als Vater bezeichnet. Der Vater kann zwei Söhne haben. Ein Sohn ist dabei alphabetisch kleiner, der andere alphabetisch größer als der Vater. Entsprechend kann jeder Sohn wieder Vater von zwei eigenen Söhnen sein. Beim Suchen hangelt sich Mainfile II dann durch den so erstellten Binärbaum. Selbstverständlich ist es bei diesem Anordungsprinzip nicht sinnvoll, Datensätze alphabetisch geordnet einzugeben, der Suchbaum würde entarten, da immer nur »große Söhne« angelegt würden.

Befehlssatz:

&CREATE legt eine Mainfile II-Datei auf der Diskette an
&MAINFILE aktiviert die Userbefehle
&PROTECT schützt ein File vor dem Löschen mit dem DOS-Befehl SCRATCH
&UNPROTECT hebt den Löschschutz auf
UAPPEND ermöglicht das Anhängen weiterer Daten an einen schon bestehenden Datensatz
UBACK sucht den alphabetischen Vorfahr
UEND beendet einen Schreibvorgang in eine Datei
UFIND sucht einen mit seinem Namen bezeichneten Datensatz
UHELP gibt den Namen, Track und Sektor des aktuellen Datensatzes aus
UINITIALIZE versetzt das Laufwerk in seinen Einschaltzustand
UJOKER definiert einen Testjoker, der beim Suchen überlesen wird
UKEY ermöglicht die Definition von bis zu acht Suchtexten
UMOVE bewirkt ein Lesen des Datenblockes durch Angabe des entsprechenden Zeigers
UNEXT sucht den alphabetischen Nachfolger
UOFF schaltet die User-Befehle ab und versetzt das Laufwerk in den Grundzustand
UPOS gestattet Direktzugriff durch Angabe von Track und Sektor
USCRATCH löscht Datensätze
UTEST testet den aktuellen Datensatz auf Name und Suchbegriffe
UVALIDATE belegt die von Mainfile II benutzten Blöcke
UWRITE legt einen neuen Datensatz an oder überschreibt einen alten

Mainfile II kann in jeder Programmiersprache, die Kommandos an die Floppy zuläßt, eingesetzt werden. An den relevanten Stellen ist nur der entsprechende Befehl einzugeben.

Fazit:

Mainfile II ist für alle fortgeschrittenen Basic-Programmierer empfehlenswert, die ein schnelles und komfortables Hilfsmittel zur Programmierung von Datenverwaltung suchen. Es muß aber an dieser Stelle vor dem nicht nur für Anfänger teilweise unverständlichen und lükkenhaften Handbuch gewarnt werden. (Um ein erstes Arbeiten mit dem Programm überhaupt erst möglich zu machen, **müssen** die Makrobefehle auf eine eigene Diskette kopiert werden.) Weiterhin sollte der Käufer bereits über umfangreiche Programmkenntnisse verfügen. Der Sinn von Mainfile II besteht darin, sich seine eigenen Datenverwaltungen zu programmieren.

(Dieter Hein/tr)

Info: Mainfile II, GES-Computer, Steinheimer Straße 22, 6450 Hanau. Unverbindliche Preisempfehlung: 98 Mark



Trends

In Deutschland wird der Jugendschutz jetzt auch bei Software sehr ernst genommen. Im August setzte die Bundesprüfstelle für jugendgefährdende Schriften sieben Spiele auf den Index. Was bedeutet dies?

Für indizierte Spiele darf nicht geworben werden. Weiterhin dürfen diese Spiele nicht mehr an Minderjährige verkauft oder ihnen zugänglich gemacht werden.

Die jetzt indizierten Spiele sind schon relativ alt, aber einmal in Gang gesetzt, wird die »schwarze Liste« wohl sehr schnell wachsen. Besonders der Versandhandel wird schwer betroffen. Wie soll man in dieser Branche Spiele anbieten, ohne werben zu dürfen?

Die Meinungen über die Entscheidung der Prüfstelle sind geteilt. Über eine Indizierung von Spielen wie zum Beispiel »Beach Head II« braucht man nicht zu diskutieren. Doch sollte sich die Indizierung so ausweiten, daß jedes Spiel, in dem man in irgendeiner Form schießen muß, auf die »schwarze Liste« kommt, hat man wohl am Ziel »vorbeigeschossen«.

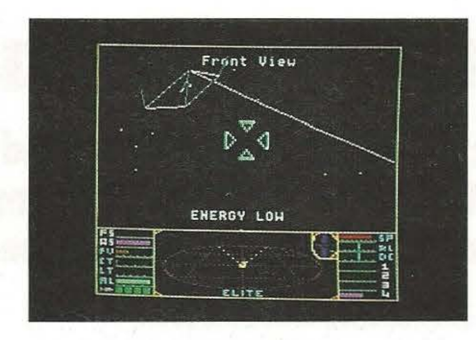
Preisbrecher

Ganz im Trend zur preiswerten Software liegen die Angebote von Quellesoft, einer neuen Abteilung des Großversandhauses Quelle. Unter diesem neuen Label gibt es Software zu sehr niedrigen Preisen. So wird ein Software-Paket angeboten, das sechs Spiele beinhaltet (unter anderen »Decathlon« und »Hunchback II«) und nur knapp 30 Mark kostet, sowie ein deutsches Actionadventure für nur 10 Mark.

Mit diesen Billig-Angeboten möchte sich nun auch Quelle einen Anteil am Softwaremarkt sichern. Der Vertrieb soll auf zwei unterschiedliche Arten erfolgen. Spiele, bei denen man vermutet, daß sie »Dauerbrenner« werden, sollen im Katalog ins Angebot kommen. Sie sind dann im Versand erhältlich. Alle anderen Programme sind nur in den einzelnen Quelle-Niederlassungen zu bekommen.

In Zukunft will die Quellesoft-Abteilung nur noch Programme vertreiben, die exklusiv bei Quelle zu bekommen sind.

Ob es auch weiterhin bei so niedrigen Preisen bleiben kann, muß sich erst erweisen. Auf jeden Fall dürfte es sich bei diesen Preisen für Raubkopierer nicht mehr lohnen, sich mit den Programmen zu beschäftigen. (rg)



50x »Elite« zu gewinnen

In der Ausgabe 9/85 stellten wir Ihnen das Spiel »Elite« vor. Hier noch einmal eine kurze Zusammenfassung: Es handelt sich bei Elite um ein dreidimensionales Weltraum-Abenteuerspiel der Spitzenklasse. Die Aufgabe besteht hauptsächlich darin, in acht Galaxien mit über 2000 verschiedenen Planeten Handel zu treiben, um möglichst viel Geld zu verdienen. Dies ist nicht so einfach, da überall Gefahren wie zum Beispiel Piraten lauern. Wenn Sie erfolgreich sind, werden Sie in Ihrer Rangstufe hoch gesetzt. Der höchste zu erreichende Rang ist »Elite«. Auf dem optimalen Weg benötigen Sie mindestens 89 Stunden, um diesen Rang zu erreichen.

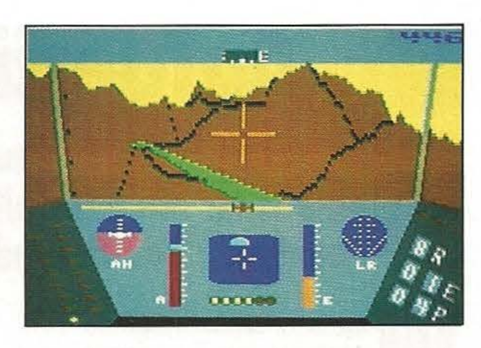
In dieser Ausgabe verlosen wir 50 Exemplare der deutschen Elite-Version, die uns vom englischen Hersteller Firebird und der deutschen Vertriebsfirma Rushware zur Verfügung gestellt wurden. Um sich an dieser Verlosung zu beteiligen, müssen Sie nur die drei unten aufgeführten Fragen beantworten.

Senden Sie Ihre Antwort an:
Markt & Technik Verlag AG
Redaktion 64'er
»Elite-Verlosung«
Hans-Pinsel-Str. 2
8013 Haar bei München

Wenn Sie sich an der Verlosung der 50 Elite-Exemplare beteiligen wollen, beantworten Sie bitte folgende Fragen. Die Antworten senden Sie bitte an die oben erwähnte Adresse.

1. Wieviele Planeten können Sie mit »Elite« anfliegen?
2. Welches ist der höchste Rang, den Sie in diesem Spiel erreichen können?
3. Wie lange benötigen Sie für die Lösung, wenn Sie den optimalen Lösungsweg nutzen?

Bitte beachten Sie den Einsendeschluß: 15. Oktober 1985. Nach diesem Termin eingehende Postkarten können leider nicht mehr berücksichtigt werden.



Rescue on Fractalus

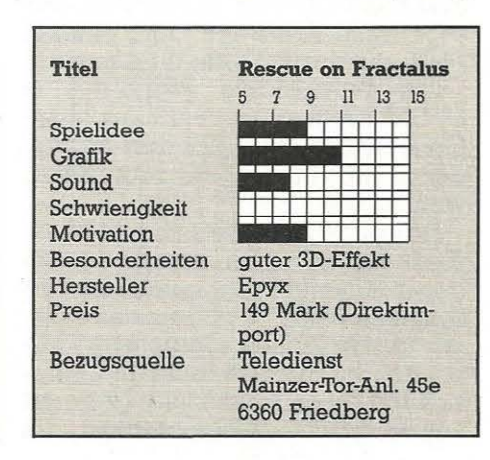
Wieder einmal gilt es, eine Invasion von Außerirdischen abzuwehren. Die dafür ausgewählten Raumpiloten haben allerdings ein kleines Problem. Die Gegner, Jaggies genannt, haben sich auf einem der schlimmsten Planeten des Universums, Fractalus, verschanzt. Fractalus ist ein zerklüfteter Gebirgsplanet mit giftiger Säure-Atmosphäre. Die Raumpiloten, die die Jaggies vernichten sollten, sind bei ihrem Einsatz abgeschossen worden. Sie warten auf der Planetenoberfläche auf ihre Rettung. Und ohne die Raumpiloten sind die Chancen eines Siegs gleich Null. Soweit zur Rahmenhandlung.

Hier setzt nun der Spieler ein. Mit einem Spezialgleiter soll er die Piloten einsammeln und im Mutterschiff abliefern. Daß die Jaggies dabei nicht tatenlos zusehen, ist wohl klar. So erwarten den Spieler Laser-Gefechtsstationen, fliegende Bomben und als Piloten getarnte Jaggies.

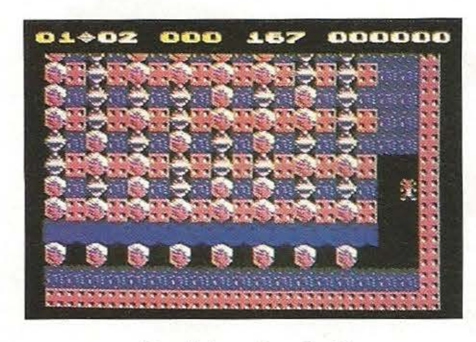
Die Ausführung des Spiels ist rundum gelungen. Eine gute, aber etwas langsame 3D-Grafik zeigt die Gebirgslandschaft, Angreifer und hervorragend animierte Piloten. Die Soundeffekte sind spärlich, aber sehr gut. Die Auftaktmusik ist guter Durchschnitt.

Rescue on Fractalus ist ein gut gemachtes Action-Spiel mit neuen Ideen und guten Effekten.

Ein wichtiger Hinweis: Dieses Spiel läuft noch nicht im C 64-Modus des C 128! (bs/rg)



Titel	Rescue on Fractalus
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	guter 3D-Effekt
Hersteller	Epyx
Preis	149 Mark (Direktimport)
Bezugsquelle	Teledienst Mainzer-Tor-Anl. 45e 6360 Friedberg



Boulder Dash II

Boulder Dash-Fans dürfen sich freuen: MicroFun hat endlich ein Folgespiel produziert. Ähnlich wie der Championship Loderunner für Loderunner-Fans, ist Boulder Dash II ebenso speziell für Personen gedacht, die den ersten Teil schon ausgiebig gespielt haben. Die Levels wurden noch wesentlich aufwendiger gemacht als die des ersten Teils. Was bedeutet: Mehr denken, schneller spielen und weniger Fehler machen.

Für diejenigen, die Boulder Dash noch nicht kennen: Ziel des Spiels ist es, Juwelen aufzusammeln und den Level lebend zu verlassen. Behindert wird man von fallenden Steinen, explodierenden Schmetterlingen (die nach der Explosion zu Juwelen werden), pulsierenden Vierecken (die einen Verfolgungsdrang zu haben scheinen) und noch einigen anderen Dingen.

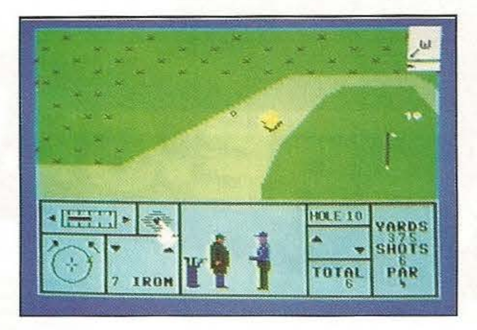
Im Vergleich zum ersten Teil hat Boulder Dash II einige neue Elemente, zum Beispiel Wände, die nach links und rechts expandieren, bis sie an den Rand oder ein Hindernis stoßen. In Boulder Dash II gibt es Levels, in denen nur ganz bestimmte Lösungssysteme funktionieren. Das soll nicht bedeuten, daß man nicht mehrere Lösungswege benutzen kann. Aber zuerst muß man eben ein ganz bestimmtes System herausfinden, mit dessen Hilfe man sich dann die Lösung erarbeitet.

Ein Muß für Boulder Dash-Fans!

(M. Kohlen/rg)



Titel	Boulder Dash II
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	—
Hersteller	First Star, Monolith
Preis	ca. 39 Mark (Kass.)
Bezugsquelle	Rushware



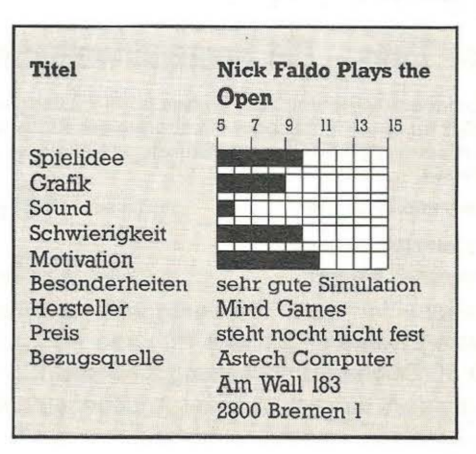
Nick Faldo Plays the Open

Nick Faldo heißt einer der weltbesten Golfspieler. Sie als Nick Faldo müssen nun zeigen, was ein guter Golfspieler kann. Als Kurs wurde der »Royal St. George's Golf Club« für das Spiel übernommen.

»Nick Faldo Plays the Open« ist ein gutes Golfspiel. Alle Schwierigkeiten der Originalgolfstrecke sind in dies Spiel eingebaut. Die Windstärke ist nicht nur miteinberechnet, sondern wechselt ständig und ist wirklichkeitsgetreu. Wählbar sind die verschiedenen Schläger, die Stärke des Schlags, die Richtung und das Loch, das man anzuspielen wünscht. Hat man das alles ausgewählt, so geht es an den Schlag selbst: Man sieht den Caddy den entsprechenden Golfschläger reichen, und der Spieler schlägt ab. Während der Ball fliegt, scrollt das Spielfeld mit. Das Spielfeld ist etwa 900 Bildschirme groß. Auf Wunsch kann natürlich eine Gesamtübersicht angesehen werden, die dann den gesamten Golfplatz auf dem Bildschirm darstellt.

Trotz der vielen Auswahlmöglichkeiten ist keine Tastatursteuerung notwendig, denn durch ein gutes Auswahlmenü kann man alle Optionen mit dem Joystick einstellen.

»Nick Faldo Plays the Open« ist ein Spiel, das auch über längere Zeit hinweg Spaß machen kann. Die Grafik ist ansprechend, auf den spärlichen Sound braucht man angesichts des interessanten Spiels eigentlich nicht zu achten. (M. Kohlen/rg)



Titel	Nick Faldo Plays the Open
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	sehr gute Simulation
Hersteller	Mind Games
Preis	steht nocht nicht fest
Bezugsquelle	Astech Computer Am Wall 183 2800 Bremen 1



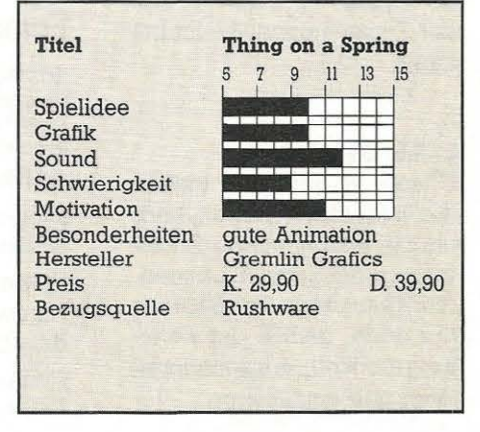
Thing on a Spring

Es ist mal wieder ein Held im Kampf gegen das Böse unterwegs. Der Bösewicht hat alle Schätze der Welt in seine unterirdische Fabrik gehext. Unserem Helden ist es gelungen, in diese Fabrik einzudringen. Der Held ist übrigens kein Mensch, sondern ein »Thing on a Spring«, ein »Ding auf einer Feder«. Sein Ziel ist natürlich die Vernichtung des Bösewichts. Auf dem Weg zu dieser Heldentat müssen insgesamt neun Puzzleteile eingesammelt werden. Zusammengesetzt geben sie einen Hinweis und das »Ding auf einer Feder« kann die versteckten Schätze wieder zusammentragen. Mit Hilfe der einzelnen Schätze erlangt er verschiedene Fähigkeiten. Der erste erlaubt das Benutzen der Lifte, mit einem weiteren kann er die Teleporter ausnutzen.

Zum Hüpfen braucht das Ding Öl, um seine Feder zu schmieren. In jedem Raum ist ein Ölbehälter zu finden. Doch er muß sich das Öl gut einteilen, um sein Ziel zu erreichen.

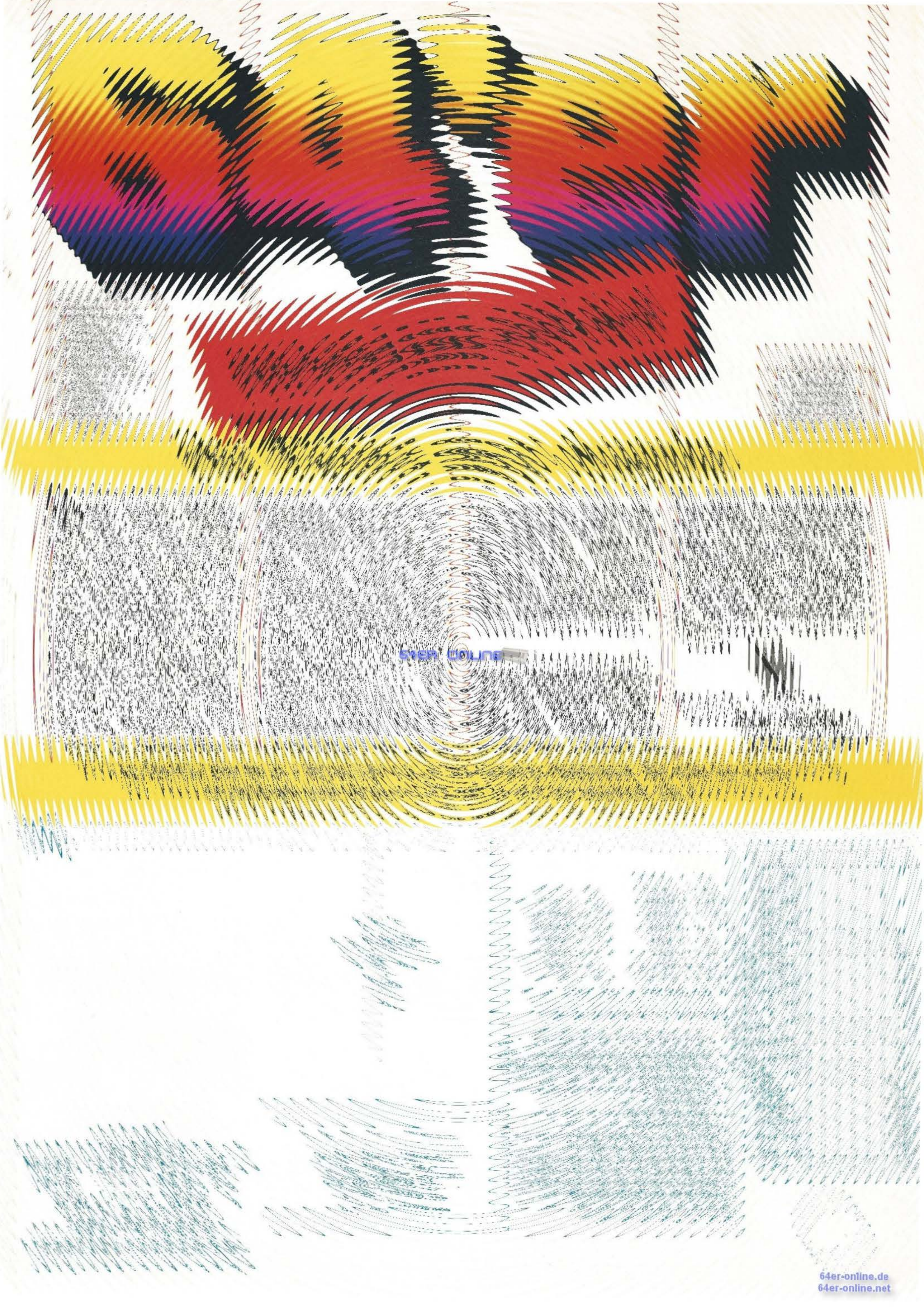
Auf dem Weg zum Bösewicht bleiben dem Spieler zwei Alternativen: erst denken, dann handeln oder einfach ausprobieren. Aber meist kommt man mit etwas Überlegen schneller ans Ziel.

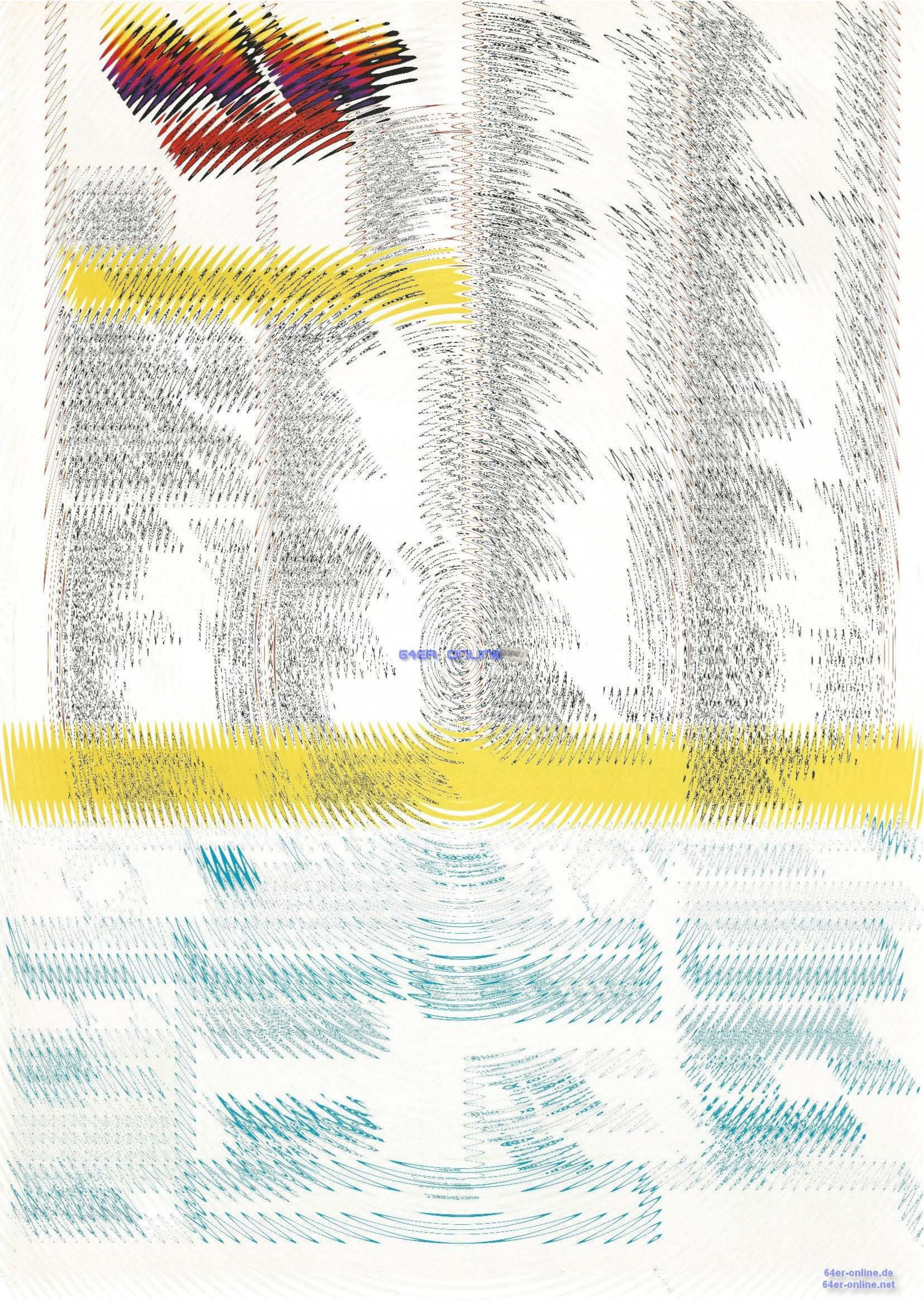
Die schöne Grafik, verbunden mit einem unvergleichlichen Sound, machen aus diesem Spiel ein Erlebnis. Thing on a Spring ist nicht einfach nur ein Hüpfspiel, es ist auch ein logisches Puzzle, das den Spieler zum Denken zwingt. (Manfred Kohlen/rg)



Titel	Thing on a Spring
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	gute Animation
Hersteller	Gremlin Grafics
Preis	K. 29,90 D. 39,90
Bezugsquelle	Rushware






64ER ONLINE

Directory-Manipulationen II

Wollen Sie nicht, daß jemand unbefugt Ihre Programmsammlung benutzt oder kopiert? Schützen Sie doch einfach Ihr Directory! Wir zeigen Ihnen, wie man's macht.

In der 64'er, Ausgabe 6/85, berichteten wir zum ersten Mal über Manipulationen an Directories. Die Resonanz auf diesen Artikel war sehr groß. Es kamen sowohl neue Tips bei uns an, wie auch Verfahren zur Überwindung der schon vorgestellten Manipulationen. Beides wollen wir Ihnen nicht vorenthalten. Deswegen also einiges mehr über Directories und deren Manipulation. Alles, was Sie an Handwerkszeug benötigen, ist ein Diskettenmonitor wie beispielsweise den aus der 64'er, Ausgabe 8/85.

Mit dem Fragezeichen überlisten

Sehr ausführlich haben wir über Tricks berichtet, die mit dem Gänsefüßchen zusammenhängen. Ein Gänsefüßchen im Filenamen sorgt für Verwirrung, weil es sich ja nicht direkt mit dem LOAD-Befehl eingeben läßt. Floppy-Experte Karsten Schramm machte uns aber auf das Fragezeichen, das zweite, selten benutzte Jokerzeichen der 1541 aufmerksam. Will man ein mit Gänsefüßchen im Filenamen gesichertes Programm laden, dann benutzt man beim LOAD-Befehl einfach das Fragezeichen. Anstelle von »LOAD "64"CHR$(34)"ER",8« funktioniert also auch »LOAD "64?ER",8«. Ähnlich kann man natürlich vorgehen, wenn Grafikzeichen oder Steuercodes in den Filenamen untergebracht worden sind. Ein bekanntes Beispiel sind die auf Diskette gespeicherten Bilder des Koala-Painter mit dem reversen Pik im Filenamen. Auch hier leistet das Fragezeichen gute Dienste.

Endlose Directories

Die Ladezeit für ein Directory beträgt normalerweise nur einige Sekunden. Das ändert sich schlagartig, wenn man ein Directory unendlich lang macht. Wer ein solches endloses Directory mit dem LOAD-Befehl laden will und nicht nach mehreren Minuten entnervt die STOP-Taste drückt, der wird mit netten optischen Effekten und einem Systemabsturz belohnt. Spätestens dann, wenn der $D-Bereich (I/O-Bausteine, VIC, etc.) überschrieben wird, spielt Ihr Computer nicht mehr mit. Wer den Ladevorgang unterbricht, kann mit dem normalen LIST-Befehl das Directory nicht lesen, da die Basic-Zeiger nicht richtig gesetzt werden. Ein Maschinensprachemonitor oder ein RENEW (OLD) können die Filenamen aber sichtbar machen. Trotzdem ist der Verblüffungseffekt groß, wenn ein einfacher LOAD-Befehl, den man jeden Tag benutzt, zum Systemabsturz führt.

Doch wie erzeugt man ein endloses Directory? Benötigt wird nichts weiter als ein einfacher Disk-Monitor. Wenn Sie das Directory auf der Spur 18 mitverfolgen, stellen Sie fest, daß die beiden ersten Bytes des letzten Directoryblocks $00 und $FF lauten.

Ändern Sie einfach diese beiden Bytes auf »12 01« (hexadezimal). Der letzte Directoryblock zeigt nun auf den ersten Directoryblock. Das Directory befindet sich jetzt sozusagen in einer Endlosschleife: Ist es zu Ende, geht es gleich wieder von vorne los. Eine recht amüsante Sache, die übrigens auch bei professioneller Software häufig zu finden ist.

Ein Directory zieht um

Beim folgenden Trick könnte man auch von »multiplen« (mehrfachen) Directories sprechen, der Begriff »Umzug« ist allerdings viel anschaulicher. Den Effekt zu beschreiben, der beim Listen von umgezogenen Directories auftritt, ist praktisch unmöglich, da sich ein umgezogenes Directory durch fast nichts von einem normalen unterscheidet. Versucht man aber, eines der weiter hinten gelegenen Programme zu laden, beispielsweise das zehnte, dann passiert alles mögliche. Vom »FILE NOT FOUND ERROR« über den »OUT OF MEMORY ERROR« bis hin zum Systemabsturz. Allerdings lassen sich die ersten acht Files ganz normal laden und starten. Diese acht Files laden nun Programmteile nach, die entweder nicht von Hand geladen werden können, weil dann oben beschriebene Effekte eintreten, oder gar nicht im Directory stehen! Was ist passiert? Ein Blick mit dem Diskettenmonitor offenbart folgendes: Der erste Directoryblock (18,1) weist nicht auf den normalerweise zweiten Directoryblock (18,4), sondern auf einen anderen, beispielsweise 18,5. Trotzdem steht in 18,4 ein vernünftiger Teil des Directorys. Dieser wird normalerweise aber niemals gelesen. Das Ladeprogramm, das auf diese Teile des Directorys zugreifen will, muß vor dem Zugriff den Zeiger in 18,1 von 18,5 auf 18,4 ändern. Dann ist das Directory so, wie es sein sollte. Nach dem Zugriff wird der Zeiger dann wieder sofort auf das Dummy-Directory, das keinerlei Funktion außer der Verwirrung hat, zurückgestellt. Es existieren also zwei unterschiedliche Directories auf der Diskette, wobei das »echte« immer nur nach einer Vorbehandlung der Diskette erreichbar ist.

Wie man so ein Dummy-Directory erstellt? Nichts einfacher als das: Kopieren Sie mit einem Disketten-Monitor Directory-Blöcke von anderen Disketten auf freie Blöcke der zu schützenden Diskette und gleichen Sie die Zeiger an. Der Begriff »Umzug« ist deswegen treffend, weil man sich ja nicht nur auf die Spur 18 beschränken muß: Der Zeiger des ersten Blockes kann beispielsweise auf 1,1 zeigen, das Directory (oder der Dummy) geht also auf der Spur 1 weiter, wo man es nicht vermutet und somit auch mit einem Diskmonitor nicht so leicht findet (Wer sucht schon alle 683 Blöcke einer Diskette nach einem Directory ab?).

Wer jetzt besonders gemein zu seinen lieben Mitmenschen sein will, der kann das Directory zum Laufwerks-Killer umgestalten. Man kombiniert dazu den Umzugtrick mit dem des endlosen Directorys: Der erste Directoryblock verweist auf 1,1, dieser verweist auf 35,1 und der wieder zurück auf 1,1. Die Folge: Beim Ladeversuch des Directories

rast der Schreib-Lese-Kopf wie wild zwischen den Spuren 1 und 35 hin und her. Bis zu über zweihundert Mal kann das gehen, betätigt man vorher nicht die STOP-Taste. Danach erhält man dann wieder den schon bekannten Systemabsturz. Warnung! Mit dieser Methode kann man die Laufwerksmechanik überlasten und somit beschädigen!

Geheimnisvolles Leerzeichen

Mit einem Leerzeichen kann man ganz verrückte Sachen anstellen. Es geht hier allerdings nicht um das Leerzeichen, das man bei Druck auf die SPACE-Taste erhält, sondern um seinen nahen Verwandten: SHIFT-SPACE. Optisch lassen sich die beiden nicht auseinanderhalten. Nur durch ihre ASCII-Codes, 32 für das normale und 160 für SHIFT-SPACE, unterscheiden sie sich. Wenn es aber um Directories geht, dann erhält SHIFT-SPACE eine tragende Bedeutung, die es weit über das normale Leerzeichen hinaushebt. Um sich die fatalen Wirkungen von SHIFT-SPACE vor Augen zu führen, sehen Sie sich doch einmal das Directory in Bild 1 an. Das sieht alles so bekannt aus, werden Sie sagen. Richtig: Ähnliche Effekte lassen sich auch mit einem Gänsefüßchen als erstem Buchstaben im Filenamen erzielen. Dummerweise lassen sich diese Programme aber nicht mit »LOAD CHR$(34)"name",8« laden. Und sogar das Fragezeichen als Ersatz für den ersten Buchstaben versagt! Was geht hier vor?

Ein Blick mit einem Disk-Monitor offenbart das Geheimnis: Alle Filenamen beginnen mit SHIFT-SPACE. Normalerweise werden Filenamen, die kürzer als 16 Buchstaben sind, mit SHIFT-SPACE aufgefüllt. Findet das DOS beim Übertragen des Directory nun ein SHIFT-SPACE, denkt es, daß der Filename somit zu Ende ist und sendet ein Gänsefüßchen an den Computer. Nachfolgende Zeichen werden aber nicht ignoriert. Will man so gesicherte Files laden, dann muß der Befehl »LOAD" SHIFT-SPACE Filenamenrest",8« lauten. Das Fragezeichen als Joker arbeitet in diesem Fall nicht einwandfrei.

Der Trick funktioniert nicht nur, wenn SHIFT-SPACE der erste Buchstabe ist. Taucht es allerdings mitten im Filenamen auf, dann ist dieser tatsächlich an der entsprechenden Stelle zu Ende. Auf diesem Trick beruhen auch Directories, an deren Filenamen ein »,8,1« angehängt wurde. Dort folgen die entsprechenden ASCII-Codes einem SHIFT-SPACE.

Was man mit dem geheimnisvollen Leerzeichen noch so alles anstellen kann, sollten Sie mit einem Disk-Monitor mal selbst ausprobieren. Nur eines sei noch gesagt: Hinter SHIFT-SPACE im Directory werden Steuercodes nicht mehr in Grafikzeichen umgesetzt, sondern ausgeführt.

Steuercodes im Filenamen

Manche Leser wollten nach dem ersten Teil wissen, was man mit Steuercodes im Filenamen anfangen kann. Nach einem Gänsefüßchen oder SHIFT-SPACE im Filenamen werden alle gesendeten Steuercodes, deren ASCII-Werte unter 128 liegen, ausgeführt. Hier ein paar Beispiele:

— CHR$(18) — REVERS ON
— CHR$(19) — HOME
— CHR$(5) — Weiß
— CHR$(28) — Rot
— CHR$(30) — Grün
— CHR$(31) — Dunkelblau
— CHR$(20) — DELETE

Ein Anwendungsbeispiel: Ist ein Filename nur sieben Buchstaben lang, dann kann man ihn wirkungsvoll im Listing verschwinden lassen, wenn hinter den sieben Zeichen des Filenamens ein SHIFT-SPACE und dann acht DELETE-Codes folgen.

Weiterhin lassen sich im Filenamen per Disk-Monitor die beiden verbotenen Jokerzeichen »*« und »?« einbauen. Ihre ASCII-Codes sind dezimal 42 und 63.

So, damit dürften wir jetzt fast alle Möglichkeiten der Directory-Manipulation ausgelotet haben. Und gleich damit verbunden eine Bitte: Senden Sie uns keinesfalls Programme, die auf diese oder andere Arten geschützt sind. Wir können nur Programme bearbeiten, die LIST-fähig sind, sowohl Directory als auch Programm. (bs)

```
0 [illegible]
12    ""64'ER          .   PRG
0     ""DAS MAGAZIN FUR  PRG
63483  ""COMPUTER-FANS     USR
256   ""[illegible]
568 BLOCKS FREE.

READY.
```

Bild 1. So kann ein Directory aussehen, das mit den hier beschriebenen Verfahren manipuliert wurde.

Fortsetzung von Seite 153

Befehle, um damit optimal arbeiten zu können. Zusätzliche Treibersoftware wird damit überflüssig.

10. System

Mit dem System-Package können eine Vielzahl von speziellen Eigenschaften des C 64-Systems angesprochen werden.

11. Font

Mit einfachen Befehlen kann der Zeichensatz des C 64 geändert, geladen oder gespeichert werden. Es ist möglich, einen speziellen Zeichensatz fest mit einem Comal-Programm zu verbinden. Wird dieses Programm gespeichert, so wird auch der zugehörige Zeichensatz ohne weiteres Zutun automatisch mit abgespeichert. Bei jedem Laden eines derart präparierten Programms steht dieser Zeichensatz dann wieder zur Verfügung.

Fazit

Die hervorragenden Eigenschaften dieser neuen Programmiersprache lassen auf eine Durchsetzung auf breiter Front hoffen. Comal ist wie kaum eine andere Sprache dazu geeignet, strukturiertes und übersichtliches Programmieren interaktiv, im Dialog mit dem Computer zu lernen. Comal nimmt dem Benutzer im Gegensatz zu anderen Sprachen so leicht nichts übel, im Gegenteil: offensichtliche Fehleingaben werden vom System automatisch korrigiert. Der eingebaute Syntax-Check bereits bei der Eingabe von Programmzeilen, ist nicht nur für den Anfänger eine große Hilfe. Ausführliche Fehlermeldungen lassen den Benutzer nicht im unklaren, was er falsch gemacht hat. Die hohe Ausführungsgeschwindigkeit von Comal-Programmen ermöglicht Anwendungen, die beim C 64 sonst nur durch Maschinensprache zu realisieren sind. Der Komfort bei der Programmierung in Comal ist für C 64-Verhältnisse so enorm, daß man sich an einen anderen Computer versetzt glaubt. Alle, wirklich alle, Unzulänglichkeiten des C 64 sind durch Comal 80 behoben. Und wenn jemand doch noch etwas finden sollte — kein Problem, denn Comal ist durch das Package-Konzept flexibel genug, vom Anwender fast beliebig erweitert werden zu können. Mit einem Wort: Comal ist nicht eine, sondern die Programmiersprache für den C 64.

(S. Bauer/ev)

Comal 80 gibt es derzeit nur als Steckmodul für den C 64 bei Commodore Data a/s, Bjerrevej 67, DK-8700 Horsens, Dänemark. Preis zirka 1000 DKr. (etwa 300 Mark).



Besser Lernen mit dem Computer

Die Lernsoftware wird immer ausgefeilter. Doch kann sie einen Lehrer ersetzen, oder ist sie nur ein Hilfsmittel zum Lernen?

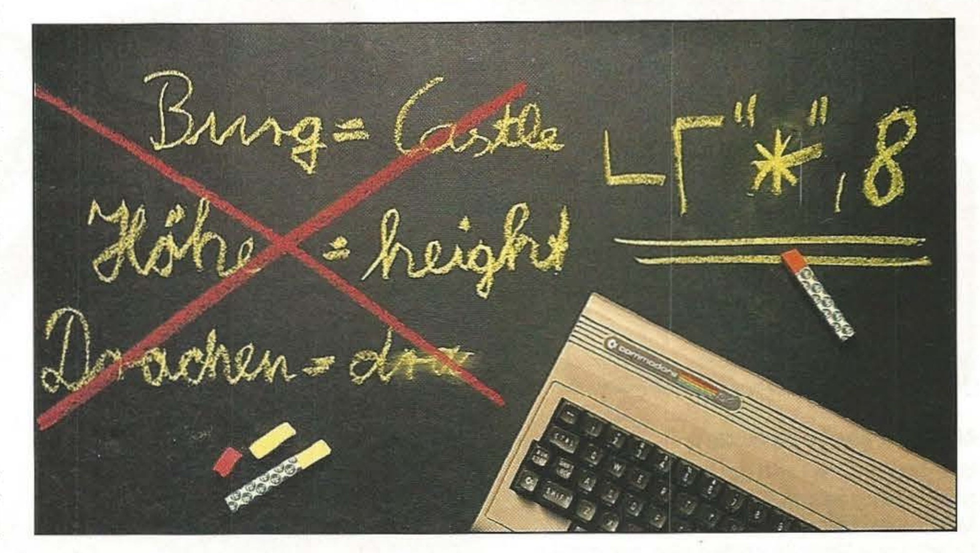
In der heutigen Arbeitswelt kann auf den Computer kaum noch verzichtet werden. Prognosen sagen voraus, daß in der nächsten Zukunft mindestens jeder zweite Arbeitnehmer an seinen Arbeitsplatz mit dem Computer zu tun hat. Die Schulen begannen bereits vor einigen Jahren sich auf diesen Trend einzustellen. Informatikkurse wurden angeboten und die ersten Computer angeschafft. Der Computer hielt Einzug in die Schulen. Doch vom Unterricht über den Computer ist es nicht weit zum Unterricht mit dem Computer.

Schule, Computer und Lernsoftware

Die »Geschichte« der Lernsoftware begann mit dem Erscheinen der Homecomputer. Bald danach kamen die ersten einfachen Vokabel-Abfrageprogramme. Diese übernahmen nur das reine Abfragen von Vokabeln und ließen in den meisten Fällen keine Doppelbedeutung von Begriffen zu. Mit dem schnell wachsenden Homecomputer-Markt gewann die Lernsoftware immer mehr an Bedeutung. Die Programme wurden verbessert, auf bestimmte Anwendergruppen zugeschnitten und auf viele Lernbereiche ausgeweitet.

Heute kann der Lernwillige auf ein breit gestreutes Lernsoftware-Angebot zurückgreifen (siehe auch unsere Marktübersicht in dieser Ausgabe). Viele der Programme sind von ihrer Themenauswahl auf Schulbücher zugeschnitten. Sie werden von den Schülern meist zu Hause genutzt, um ihre Leistungen in der Schule zu verbessern. In den Schulen selbst wird von dieser Möglichkeit jedoch seltener Gebrauch gemacht. Doch immerhin empfehlen schon einige Lehrer ihren Schülern, die einen Computer besitzen, von dem »Nachhilfelehrer Computer« Gebrauch zu machen.

Zu der Lernsoftware sind in letzter Zeit Simulationsprogramme dazugekommen. Wie schon aus dem Namen ersichtlich ist, werden hier mit dem Computer bestimmte Situationen ausgeführt. Auf Großcomputern sind Simulationen schon lange im Gebrauch. Sie werden zum Beispiel in der Pilotenausbildung und im Flugzeugbau eingesetzt. Für den C 64 bewegen sich diese allerdings noch auf der spielerischen Ebene, wie zum Beispiel »Flightsimulator II« und »Injured Engine«.



Der Trend

Die Entwicklung der Lernprogramme in der letzten Zeit weist einen deutlichen Weg. Immer besser durchdachte und für bestimmte Anwendergruppen ausgelegte Programme erscheinen. Die grafischen Fähigkeiten des Computers werden gezielt zur Unterstützung des Lernerfolges eingesetzt. So lassen Vokabelprogramme, wie »Superlearning« verschiedene Anfrageformen zu, die vom Lückentext-Diktat bis hin zu Silbenrätseln reichen.

Man kann in puncto Lernsoftware optimistisch in die Zukunft sehen, dann das Lernen wird, Dank dem Computer, immer mehr zum Vergnügen. Nur eines, das kann der Computer Ihnen nicht abnehmen. Lernen, müssen Sie immer noch selbst. (rg)

Kurse: Einstieg in die

Der Weg, seinen Computer kennenzulernen, ist

Die verschiedensten Arten von Kursen werden angeboten. Diese reichen von Büchern bis hin zu Computer-Camps. Eine allgemeingültige Antwort, welche Art von Kursen besonders geeignet ist, kann nicht so einfach gegeben werden.

Die Bücher

Im deutschsprachigen Raum wird eine nahezu unüberschaubare Menge von Büchern zum Thema Computer angeboten. Viele hiervon verstehen sich als Lehrbücher. Bei Büchern für Anfänger wird meist anhand von kleinen Beispielprogrammen auf bestimmte Fähigkeiten und Eigenschaften des jeweiligen Computers eingegangen. Aber leider läßt die Qualität vieler dieser Beispiele oft zu Wünschen übrig. Der Erfolg, auf diesem Weg den Computer kennenzulernen, ist oft gleich Null. Für Fortgeschrittene finden sich im Buchangebot allerdings einige gute Nachschlagwerke.

Computerkurs auf Diskette

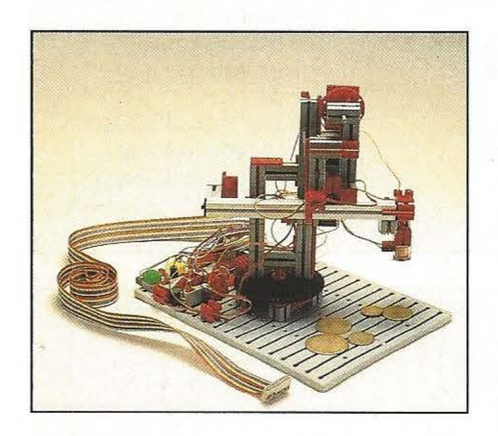
Computerkurse auf Diskette ist eine bisher recht wenig genutzte Art, den Computer kennenzulernen. Uns sind nur drei deutsche Anbieter bekannt, die solche Kurse für Anfänger anbieten. Sie sind gut geeignet, um den ersten Kontakt mit dem Computer herzustellen.

Computerunterricht an Schulen

Seit einigen Jahren wird an den Schulen Informatik unterrichtet.

Roboter selbst gebaut

Mit einem Roboterbausatz können Sie sich jetzt Ihre eigenen Roboter bauen. Wie Sie dies machen, zeigt unser Bericht.



Die in der Industrie eingesetzten Roboter sind oft sehr teuer und ihre Programmierung ist meist schwer zu verstehen. Fischertechnik Computing bietet jetzt einen Roboterbausatz für verschiedene Heimcomputer an. Anhand ihres Aufbaus und ihrer Programmierung lernt man ihre »großen Brüder« in der Industrie zu verstehen. Öffnet man die Verpackung, bekommt man zuerst die von normalen Fischertechnik-Baukästen bekannten Quaderbausteine zu Gesicht. Aus diesen Einzelteilen setzen sich im wesentlichen die verschiedenen Geräte zusammen, die man aus diesem Kasten zusammenbauen kann. Zwei Motoren bilden dabei das Herzstück der einzelnen Roboter.

Weiterhin benötigt man ein Steuerinterface für den C 64, dem auch die Programmdiskette beigefügt ist. Für das Interface braucht man aufgrund des großen Strombedarfs noch ein Netzteil, das das Interface mit der nötigen Zusatzspannung versorgt. Die beiligende Anleitung liefert Anregungen für den Bau von zehn verschiedenen Geräten, die alle über den Computer angesteuert werden. Das sind unter anderem eine Ampelanlage, eine Sortieranlage, ein Sortier-Roboter, ein Roboter, der die Bausteine der Türme von Hanoi transportiert, ein Teach-in-Robot oder ein Plotter.

Der Aufbau muß in der Anleitung anhand von Bildern nachvollzogen werden, was jedoch manchmal etwas Phantasie und ein gutes Sehvermögen erfordert. Dies liegt vor allem daran, daß die einzelnen Aufbauschritte teilweise zu grob unterteilt sind. Hat man das Gerät anhand des Schaltplans verdrahtet und das Interface angeschlossen, so muß nur noch das entsprechende Programm in den Computer geladen werden, damit der Roboter »Leben« eingehaucht bekommt. Die dazugehörige Software ist in Basic geschrieben und baut auf einem Maschinenprogramm — für die Bedienung des Interface — auf. Die Programmierung ist aber sehr einfach, so daß man sich schnell mit ihr vertraut machen kann.

Natürlich ist es auch möglich eigene Geräte zu entwerfen, zu bauen und deren Ansteuerung zu programmieren. Sollten dazu die vorhandenen Bausteine nicht ausreichen, so kann man in Ergänzung auf normale Fischertechnikkästen zurückgreifen.

Wie ist es aber um den pädagogischen Wert dieser Bausätze bestellt! Nun, gerade dadurch, daß man die einzelnen Geräte selbst zusammenbauen muß, fällt es einem leicht, die Arbeitsweise zu verstehen. So ist es zum Beispiel bei der Bewegungssteuerung nicht nur mit dem einfachen Ein-/ Ausschalten der Motoren getan ist. Schließlich muß sich der Roboter auch »Orientieren« können, was über eingebaute Schalter oder Potentiometer möglich ist. Das Interface wertet diese Signale aus und sendet sie an den Computer. Dieser kann dann, wiederum über das angeschlossene Interface, entsprechend reagieren.

Wir hatten das Gerät aufgebaut, das die Bausteine für die Türme von Hanoi bearbeitet. Mit Hilfe eines Potentiometers in der Längsachse kann der C 64 die Position abfragen, damit ist eine mehr oder weniger exakte Positionierung des Roboters möglich. Leichte Ungenauigkeiten machen sich bei diesem Roboter nicht allzu negativ bemerkbar. Beim Plotter allerdings können diese Ungenauigkeiten zu Abweichungen führen, die nicht mehr zu akzeptieren sind.

Dieser Bausatz wurde hauptsächlich für diejenigen zusammengestellt, die mit ihrem C 64 in die Roboterprogrammierung hineinschnüffeln wollen. Mit einem Gesamtpreis von zirka 600 Mark für Grundkasten, Interface und Netzteil, dürfte der Roboterbausatz wohl deutlich über dem liegen, was ein privater Anwender für so ein »Spielzeug« ausgeben kann. Sinnvoll könnten diese Kästen für den Informatikunterricht der Schulen sein. Hier wäre allerdings das Engagement der Lehrer gefragt, die diesen Unterricht sinnvoll aufbauen müssen.

(Christoph Sauer/rg)

Info: Fischer-Werke, Artur Fischer GmbH & Co. KG, Weinhalde 16—18, 7244 Tumlingen/Waldachtal.

Welt der Computer?

weit. Eine Hilfe sollen hier Computerkurse bieten.

Dieser Unterricht bietet einige große Vorteile, die bei den bisher aufgeführten Kursen fehlen. Hauptsächlich ist dies der direkte Kontakt zu jemanden, der bei auftretenden Problemen helfen kann und dem Fragen gestellt werden können. Doch zur Zeit gibt es an einigen Schulen noch Probleme. Zum einen mangelt es an Fachkräften, die Informatik unterrichten können. Zum anderen sind dies fehlende Geräte. Wenn für eine Klasse von 30 Schülern nur fünf Computer zur Verfügung stehen, muß der Lernerfolg stark bezweifelt werden.

Computer-Camps

Mit dem Computer Ferien machen, ist die Devise der Computer-Camps. Über einen Zeitraum von ein bis zwei Wochen wird der erste Kontakt zum Computer hergestellt. Unter fachlicher Anleitung wird man mit dem Computer vertraut gemacht und bis zur Erstellung eigener Programme geführt. Computer-Camps sind auf jugendliche Computer-Neulinge ausgelegt. Ein Computer-Camp für Erwachsene ist uns nicht bekannt.

Fazit

Den Computer-Kurs, der für jeden gleichermaßen geeignet ist, gibt es zur Zeit noch nicht. Aus dem vorhandenen Angebot muß man sich das heraussuchen, was einem für sich persönlich am geeignetsten erscheint. (rg)



Marktübersicht: Lernsoftware

Über 130 verschiedene Lernprogramme sind in Deutschland erhältlich. Diese Marktübersicht bietet Ihnen eine Auswahl- und Entscheidungshilfe.

Das Lernsoftware-Angebot in Deutschland erstreckt sich hauptsächlich auf den schulischen Bereich. Die Produkt-Palette reicht von Vokabel-Lernprogrammen bis hin zu Programmen über Enzymkinetik. Viele dieser Programme sind auf bestimmte Schulstufen und -klassen ausgelegt. Unsere Marktübersicht zeigt Ihnen die wichtigsten Merkmale der Programme und bietet so eine Auswahl- und Entscheidungshilfe für den Kauf.

Alle in der Marktübersicht angeführten Daten beruhen auf Herstellerangaben. Die Marktübersicht erhebt keinen Anspruch auf Vollständigkeit. (rg/og)

ANBIETER:
a Ariola-Soft
b Data Becker
c Ernst-Klett-Verlag
d Hagemann
e Haller-Verlag
f Happy-Software
g Homesoft
h IPPL-Kiel
i Langenscheid
j Max-Hueber-Verlag
k Ravensburger
l SM Soft-Learning
m Sybex-Verlag
n Tender Art
o Unterrichtsmethoden Bernhard Hoppius
p Westermann-Software

Anmerkung:
*) Systembasis benötigt

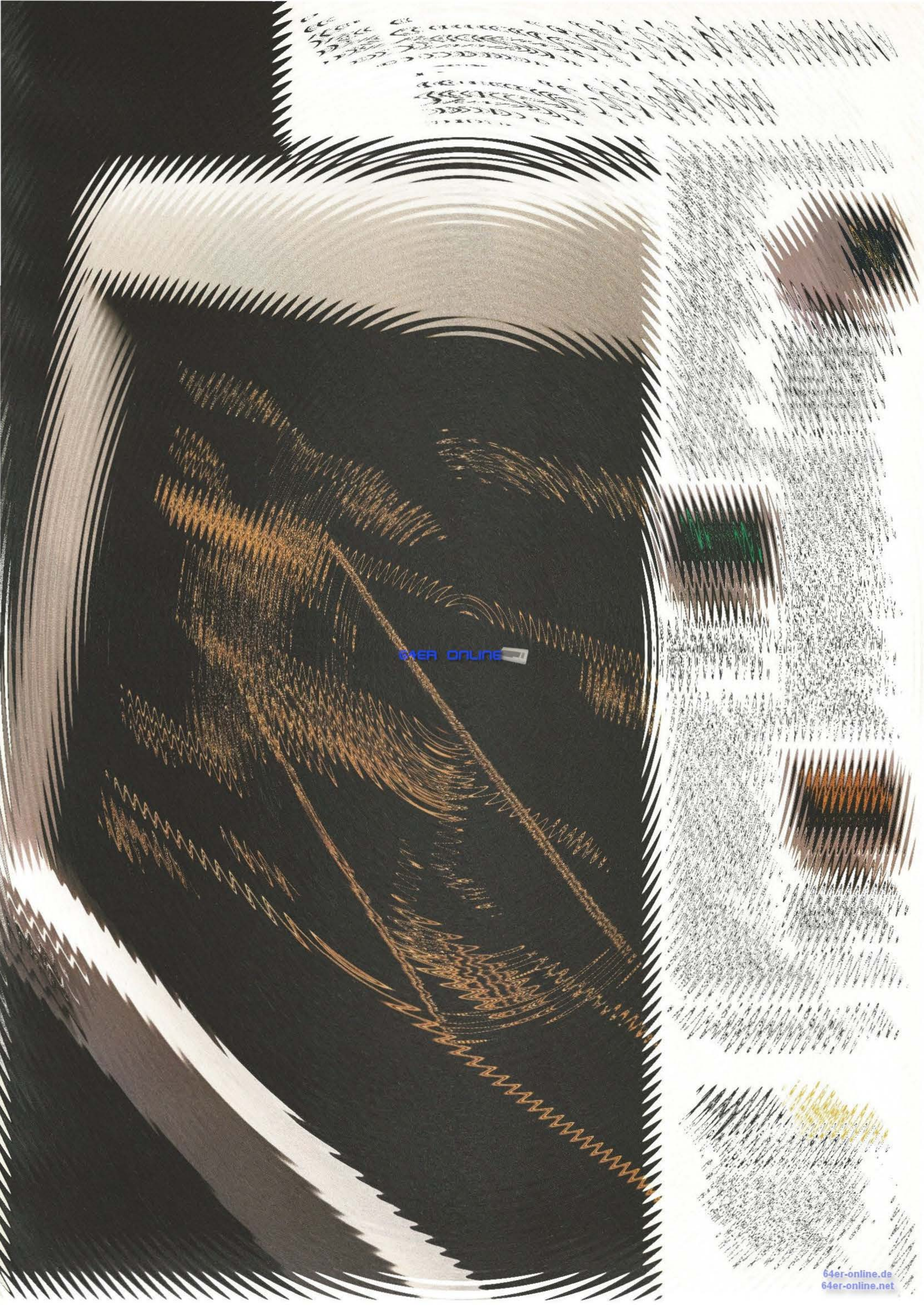
Programmname	Anbieter	Lernstoff	Altersgruppe in Jahren	a) Schultyp und Klasse b) Erwachsenenbildung	Anzahl der Lektionen	A = Arbeitsmappe B = Buch H = Begleitheft K = Audiokassette P = Programmierblock	Grafik enthalten? a=Einzelpkt. b=Blockgraf.	Softwaremedium K = Kassette D = Diskette M = Modul	Preis pro Lektion, inklusive Mehrwertsteuer
SPRACHEN									
Systembasis	a	Probelektion Esperanto	ab 14	a) — b) Ja	1	B+K	a	D	89,—
Spanisch Grundkurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Italienisch Grundkurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Englisch Grundkurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Englisch Aufbaukurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Managemant Englisch	a	Wirtschafts-englisch	ab 14	a) — b) Ja	1	B+K	a	D	198,—
Französisch Grundkurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Französisch Aufbaukurs	a		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Supervoc	a	Vokabel-Lern-Programm	ab 16	a) — b) Ja	1	B	k.A.	D	49,90
Polissez votre Francais	b	Vokabeltrainer	ab 18	a) — b) —	3	H	—	D	jeweils 49,—
Brush up your english	b	Vokabeltrainer	ab 18	a) — b) —	3	H	—	D	jeweils 49,—
Vokabeltraining, englisch	c	unregelmäßige Verben	ab 10	a) ab 3. Lernj. b) Ja	1	—	a	K D	48,— 58,—
Vokabeltraining, französisch	c		ab 10	a) ab 3. Lernj. b) Ja	1	—	a	K D	48,— 58,—
Vokabeltraining, italienisch	c		ab 10	a) ab 3. Lernj. b) —	1	—	a	K D	48,— 58,—
Vokabeltraining, spanisch	c		ab 10	a) ab 3. Lernj. b) —	1	—	a	K D	48,— 58,—
Rechtschreiben mit Köpfchen	c		7—14	a) ab 3. Kl. b) —	3	—	—	D	jeweils 79,—
Lateinische Deklination	d		bis 14	a) — b) —	1	H	—	K	34,50
Lateinische Konjuationen	d		bis 14	a) — b) —	1	H	—	K	34,50
Vokabel-Trainer/ Vokabel-Writer	d		bis 14	a) — b) —	1	H	—	K D	38,50 49,—
Vokabel-Trainer mit Wortschatz »Englisch G2«	d		bis 14	a) — b) —	1	H	—	D	48,—
Englisch	e		ab 18	a) — b) Ja	1	K	b	D	98,—
Französisch	e		ab 18	a) — b) Ja	1	K	b	D	98,—
Deutsch für Ausländer	e		ab 18	a) — b) Ja	1	K	b	D	98,—

Programmname	Anbieter	Lernstoff	Altersgruppe in Jahren	a) Schultyp und Klasse b) Erwachsenenbildung	Anzahl der Lektionen	A = Arbeitsmappe B = Buch H = Begleitheft K = Audiokassette P = Programmierblock	Grafik enthalten? a=Einzelpkt. b=Blockgraf.	Softwaremedium K = Kassette D = Diskette M = Modul	Preis pro Lektion, inklusive Mehrwertsteuer
Wortschatz-Trainer Italienisch	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer Spanisch	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer Englisch	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer Französisch	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer Latein I	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer Latein II	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Wortschatz-Trainer unregelmäßige Verben Latein	f		ab 7	a) ab 5. Kl. b) Ja	1	H	a,b	D	59,–
Rechtschreibung 1 bis 3	h		ab 10	a) ab 5. Kl. b) Ja	3	—	—	2D+M 3D 2D	120,– (* 148,90 (* 99,–
Fremdwörtertraining	h	Fremdwörter-Rechtschreibung	ab 11	a) ab 6. Kl. b) Ja	1	—	—	D	49,50
Englisch-Nachhilfe 1	h	typische Fehler	ab 12	a) ab 7. Kl. b) Ja	1	—	—	D	(* 49,50
Therapie-Ausgabe, gesamtes Programm der IPPL, Kiel	h				—	—	—	10D+M	500,–
Sesam, öffne dich!	i	Engl./franz., wichtige Vokabeln	ab 11	a) 1./2. Lernj. b) —	1	H	b	B	49,–
Wörter in Aktion	i	Engl./franz., Rechtschreibung	ab 11	a) 1. Lernj. b) —	1	H	b	B	49,–
Wordmatch	i	Englisch	ab 14	a) ab 2./3. Lernj. b) —	1	H	b	B	49,–
Teste Deine Verben	i	Französisch	ab 11	a) 1. Lernj. b) —	1	H	b	B	49,–
Teste Deinen Wortschatz 1 und 2	i	Englisch	ab 14	a) Anfänger/ Fortgeschrittene b) —	1	H	b	B	49,–
Wörterrennen mit System	i	Englisch, Grundwortschatz, Hauptwörter	ab 11	a) Anfänger b) —	1	H	b	B	49,–
Vorsicht, Vokabeln greifen an	i	Engl./franz. Latein	ab 11	a) Anfänger b) —	1	H	b	B	49,–
Jagd auf Vokabeln	i	Engl./franz.	ab 14	a) 3./4. Lernj. b) —	1	H	b	B	49,–
Achtung Vokabelfalle	i	Englisch	ab 14	a) 3./4. Lernj. b) —	1	H	b	B	49,–
Sturm auf die Burg	i	Engl./franz.	ab 11	a) 1./2. Lernj. b) —	1	H	b	B	49,–
No siempre sera asi	j	Spanisch	ab 18	a) — b) Ja	1	—	b	K	49,–
Ya viendrau tiempos mejores	j	Spanisch	ab 18	a) — b) Ja	1	—	b	K	49,–
In Europa unterwegs	j	Englisch, französisch, italienisch, spanisch	ab 18	a) — b) Ja	1	—	b	K	59,–
In Skandinavien unterwegs	j	Schwedisch, finnisch, norwegisch, dänisch	ab 18	a) — b) Ja	1	—	b	K	59,–
Take it easy	j	Englisch	ab 14	a) — b) Ja	1	—	b	K	49,–
Keep smiling	j	Englisch	ab 14	a) — b) Ja	1	—	b	K	59,–
Sur le bon chemin	j	Französisch	ab 14	a) — b) Ja	1	—	b	K	49,–

Programmname	Anbieter	Lernstoff	Altersgruppe in Jahren	a) Schultyp und Klasse b) Erwachsenenbildung	Anzahl der Lektionen	A = Arbeitsmappe B = Buch H = Begleitheft K = Audiokassette P = Programmierblock	Grafik enthalten? a=Einzelpkt. b=Blockgraf.	Softwaremedium K = Kassette D = Diskette M = Modul	Preis pro Lektion, inklusive Mehrwertsteuer
Ca y est	j	Französisch	ab 14	a) — b) Ja	1	—	b	K	59,—
Prima detto che fatto	j	Italienisch	ab 18	a) — b) Ja	1	—	b	K	49,—
Systembasis	l	Probelektion Esperanto	ab 14	a) — b) Ja	1	B+K	a	D	89,—
Spanisch Grundkurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Italienisch Grundkurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Spanisch Aufbaukurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Italienisch Aufbaukurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Englisch Grundkurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Englisch Aufbaukurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Management Englisch	l	Wirtschaftsenglisch	ab 14	a) — b) Ja	1	B+K	a	D	198,—
Französisch Grundkurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Französisch Aufbaukurs	l		ab 14	a) — b) Ja	1	B+K	a	D	198,—
Englischtrainer 64	n	Übersetzen	allg.	a) Allgemein b) Ja	1	—	b	D	45,—
Italienischtrainer 64	n	Übersetzen	allg.	a) Allgemein b) Ja	1	—	b	D	45,—
Spanischtrainer 64	n	Übersetzen	allg.	a) Allgemein b) Ja	1	—	b	D	45,—
Lateintrainer 64	n	Übersetzen	allg.	a) Allgemein b) Ja	1	—	b	D	45,—
Rechtschreiben mit Köpfchen I	o	Auslautverhärtung	7—14	a) Grundschule, Sek I b) —	1	H+A	a,b	D	79,— Arbeitsmappe 39,50
Rechtschreiben mit Köpfchen II	o	Auslautverhärtung	7—14	a) Grundschule, Sek 1 b) —	1	H+A	a,b	D	79,— Arbeismappe 39,50
Rechtschreiben mit Köpfchen III	o	Auslautverhärtung	7—14	a) Grundschule, Sek 1 b) —	1	H+A	a,b	D	79,— Arbeitsmappe 39,50
Rechtschreibung	p		7—10	a) Grundschule, 3./4. Kl., Sek 1 b) —	5	B	a,b	Teil 1, K+M oder D+M Teil 2-5 K oder D	49,— 49,— jeweils 19,—
Englische Grammatik	p		ab 16	a) Sek I b) Ja	5	B	a,b	Teil 1, D Teil 2-5, D	59,— jeweils 25,—
Französische Grammatik	p		ab 16	a) Sek I & Sek II b) Ja	5	B	a,b	Teil 1, D Teil 2/3 u. 4/5, D	59,— jeweils 25,—
Spanische Grammatik	p		ab 16	a) Sek I & Sek II b) Ja	5	B	a,b	Teil 1, D Teil 2/3 u. 4/5, D	59,— jeweils 25,—
Italienische Grammatik	p		ab 16	a) Sek I/II b) Ja	4	B	a,b	Teil 1/2, D Teil 3/4, D	59,— 25,—
Englische Vokabeln	p		ab 16	a) Sek I/II b) Ja	4	B	a,b	Rahmenprogramm, Datenteile	49,— jeweils 25,—
Französische Vokabeln	p		ab 16	a) Sek I/II b) Ja	4	B	a,b	Rahmenprogramm, Datenteile	49,— jeweils 25,—
COMPUTERKURSE									
Computer-Selbstlernkurs	a	Computerkurs für Einsteiger	ab 18	a) — b) Ja	5	B+K	a,b	D	zusammen 99,—
Computer-Selbstlernkurs	g	Computerkurs für Einsteiger	ab 18	a) — b) Ja	5	B+P	a,b	D	zusammen 99,—
Sag's der Schildkröte	k	Logo	ab 9	a) k.A. b) k.A.	1	k.A.	k.A.	D	79,—
Kindercomp	k		5—8	a) k.A. b) k.A.	1	k.A.	k.A.	M	59,—

Programmname	Anbieter	Lernstoff	Altersgruppe in Jahren	a) Schultyp und Klasse b) Erwachsenenbildung	Anzahl der Lektionen	A = Arbeitsmappe B = Buch H = Begleitheft K = Audiokassette P = Programmierblock	Grafik enthalten? a=Einzelpkt. b=Blockgraf.	Softwaremedium K = Kassette D = Diskette M = Modul	Preis pro Lektion, inklusive Mehrwertsteuer
Fratzenschneider	k	Computerbefehle und Konzentration	5—10	a) k.A. b) k.A.	1	k.A.	k.A.	M	69,—
Commodore 64 Basic-Kurs	m		ab 16	a) — b) Ja	1	B	—	K D	64,— 64,—
Commodore 64 Assembler-Kurs	m		ab 16	a) — b) Ja	1	B	—	K D	64,— 64,—
Commodore 64 Basic-Abenteuer	m		14—16	a) — b) Ja	1	B	—	K D	38,— 38,—
MATHEMATIK									
Junior Mathemat	b	Lern- und Trainingsprogramm	7—10	a) Grundschule 1.—4. Kl. b) —	1	H	a	D	69,—
Bruchrechnen einfach	c	Bruchrechnen	10—14	a) ab 6. Kl. b) —	1	—	a	K D	48,— 58,—
Algebra — Hausaufgaben leichtgemacht	c		10—16	a) 5.—11. Kl. b) —	1	—	b	D	99,—
Grundrechenarten, schriftlich	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Grundrechenarten, schriftlich	d	Kopfrechnen	bis 14	a) — b) —	1	H	a,b	K	38,50
Grundrechenarten	d		bis 14	a) — b) —	1	H	a,b	D	65,—
Bruchrechnen I	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Bruchrechnen II	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Bruchrechnen	d		bis 14	a) — b) —	1	H	a,b	D	65,—
Prozentrechnen I	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Prozentrechnen II	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Zinsrechnen	d		bis 14	a) — b) —	1	H	a,b	K	38,50
Prozent- und Zinsrechnen	d		bis 14	a) — b) —	1	H	a,b	D	69,—
Funktionen	d		bis 16	a) — b) —	1	H	a,b	K D	34,50 45,—
Kurvendiskussion	d		bis 16	a) — b) —	1	H	a,b	K D	34,50 45,—
Springteufel	k	Bruchrechnen	ab 7	a) k.A. b) Ja	1	k.A.	k.A.	M	69,—
Bilderregen	k	Zahlen	5—9	a) k.A. b) k.A.	1	k.A.	k.A.	M	69,—
SYBEX-Tutor	m	Mathematik	7—18	a) 7.—11. Kl. b) —	4	—	a	D	jeweils 49,80
Die Rechentafel — Bruchrechnen	o		ab 13	a) Grundschule, Sek I/II b) Ja	1	H	a,b	D	99,—
Geometrie 3./4. Schuljahr	p		7—10	a) Grundschule 3./4. Kl. b) —	1	B	a,b	M	49,—
Mathematik 1. Schuljahr	p		bis 7	a) Grundschule 1. Kl. b) —	1	B	a,b	M	49,—
Add. & Subtr. 2./3. Schuljahr	p		7—10	a) Grundschule 2./3. Kl. b) —	1	B	a,b	M	49,—
Mult. & Div. 2./3. Schuljahr	p		7—10	a) Grundschule 2./3. Kl. b) —	1	B	a,b	M	49,—
Add. & Subtr. 3./4. Schuljahr	p		7—10	a) Grundschule 3./4. Kl., Sek I b) —	1	B	a,b	M	49,—
Mult. & Div. 3./4. Schuljahr	p		7-10	a) Grundschule 3./4. Kl., Sek I, OS b) —	1	B	a,b	D	49,—

Programmname	Anbieter	Lernstoff	Altersgruppe in Jahren	a) Schultyp und Klasse b) Erwachsenenbildung	Anzahl der Lektionen	A = Arbeitsmappe B = Buch H = Begleitheft K = Audiokassette P = Programmierblock	Grafik enthalten? a=Einzelpkt. b=Blockgraf.	Softwaremedium K = Kassette D = Diskette M = Modul	Preis pro Lektion, inklusive Mehrwertsteuer
Schlußrechnung	p		ab 16	a) Sek II/I b) Ja	1	B	a,b	M	49,–
Prozentrechnung	p		ab 16	a) Sek II/I b) Ja	1	B	a,b	M	49,–
Zinsrechnung	p		ab 16	a) Sek I/II b) Ja	1	B	a,b	M	49,–
Bruchrechnung I	p		ab 10	a) Sek I b) Ja	1	B	a,b	D	49,–
Bruchrechnung II	p		ab 10	a) Sek I/II b) Ja	1	B	a,b	D	49,–
Schlußrechnung	p		ab 16	a) Sek I/II b) Ja	1	B	a,b	D	49,–
Prozentrechnung	p		ab 16	a) Sek I/II b) Ja	1	B	a,b	D	49,–
Zinsrechnung	p		ab 16	a) Sek I/II b) Ja	1	B	a,b	D	49,–
MUSIK									
3001 Sound Odyssee	a	Grundlagen Musik	ab 14	a) — b) Ja	1	B	a	K	59,90
Melodienschreiber	d		bis 18	a) — b) —	1	H	a,b	K	48,–
Musik-Synthesizer	d		bis 18	a) — b) —	1	H	a,b	K	38,50
Melodienschreiber-Synthesizer	d		bis 18	a) — b) —	1	H	a,b	D	69,–
Quintenzirkel und Akkorde	d		bis 18	a) — b) —	1	H	a,b	K D	38,50 49,–
NATURWISSENSCHAFTEN									
Chem. Gleichgewicht — Haber-Bosch-Verfahren	c	Chemie	16-18	a) Sek II b) —	1	B	a	D	120,–
Enzymkinetik	c	Biologie	16-18	a) Sek II b) —	1	B	a	D	120,–
Genkartierung	c	Biologie	16-18	a) Sek II b) —	1	B	a	D	120,–
Freier Fall	d	Physik	bis 16	a) — b) —	1	H	a,b	K D	34,50 45,–
Beschleunigung	d		bis 16	a) — b) —	1	H	a,b	K	38,–
Widerstand I	d		bis 16	a) — b) —	1	H	a,b	K D	34,50 45,–
Lichtbrechung	d		bis 16	a) — b) —	1	H	a,b	K D	34,50 45,–
Evolution	d	Biologie	bis 18	a) — b) —	1	H	a,b	K D	34,50 45,–
Molekularkinetik: DNA/DNS	d	Biologie	bis 18	a) — b) —	1	H	a,b	K D	38,50 49,–
Periodic table of the elements	n	Chemie	ab 7	a) 7.-12. Kl. b) Ja	1	—	—	K D	18,– 28,–
Physik	n	Mechanik, Wärmelehre, Elektronik, Akustik, Optik	ab 7	a) 7.-12. Kl. b) Ja	1	—	—	2 D	54,–
SONSTIGE									
Morsetrainer	a		ab 14	a) —, b) Ja	1	B	k.A.	D	69,40
Schreibmaschinenkurs	a		ab 16	a) —, b) Ja	1	B	k.A.	D	49,90
CNCTIX, CNC-Simulationsprogramm Drehen	c	Metallberufe	ab 16	a) Berufsschule b) Ja	1	B	a	D	750,–
Schreibmaschinenkurs	h	10-Finger-Blindschreiben	ab 11	a) ab 2. Kl. b) Ja	1	H	—	D+M D	120,– (* 49,50
Union Pacific	k	Wirtschaftl. Zusammenhänge	ab 10	a) k.A. b) Ja	1	k.A.	k.A.	D	79,–
Die Spürnasen	k	logisches Denken	ab 10	a) k.A. b) Ja	1	k.A.	k.A.	D	69,–
Geschichtenschreiber	k	Sprache	ab 8	a) k.A., b) k.A.	1	k.A.	k.A.	D	69,–
Streng geheim	k	Datenverarbeitung	ab 9	a) k.A., b) k.A.	1	k.A.	k.A.	D	69,–



In den letzten Wochen und Monaten waren sowohl für unsere Redaktionsassistenz als auch für uns Redakteure die neuesten Teilnehmerzahlen immer wieder Grund zum Staunen. Daß es den meisten nicht nur um die ausgesetzten Preise gegangen ist, zeigen die fast ausnahmslos vollständig ausgefüllten Fragebögen. Die Antworten auf den Fragebögen waren ausführlich, ehrlich und manchmal auch konstruktiv-kritisch. Einige Leser sendeten uns sogar, zusammen mit dem Fragebogen, seitenlange Briefe ein. Dies ist für uns vor allem ein Zeichen dafür, daß die Mehrzahl unserer Leser aktiv an der Gestaltung der 64'er teilnehmen möchte. Und das wird berücksichtigt, denn wir nehmen Ihre Meinung ernst. In der größten Umfrageaktionen seit Bestehen der 64'er wurden alle Ihre Wünsche, Vorstellungen und Anregungen erfaßt und statistisch ausgewertet. Es ist unvorstellbar, welche Datenmengen dabei zusammengekommen sind. Die Endauswertung ist ein über 200 Seiten (DIN A 3) starkes Dokument und mußte von einem Cyber-Großcomputer errechnet werden. Wichtigste Erkenntnis daraus ist, daß es den typischen 64'er-Leser nicht gibt. Jeder unserer Leser verwendet den Computer auf seine ganz individuelle Weise. Der Anteil derer, die mit dem Computer mehr als nur spielen wollen, ist dabei außerordentlich hoch. Wir wollen uns aber nicht in Zahlen verlieren. Eines ist auf jeden Fall sicher: Alle Ihre Vorstellungen darüber, wie die 64'er in Zukunft aussehen soll, werden gewissenhaft geprüft. Seien Sie gespannt, vielleicht erkennen Sie Ihre Meinung in den nächsten Ausgaben wieder. Einen Wunsch vieler Leser können wir bereits in dieser Ausgabe erfüllen: Mehr Hardware-Selbstbauanleitungen. Und damit auch die Leser, die nicht in der Lage sind, sich Platinen selbst zu bauen, in den Besitz unserer Veröffentlichungen kommen, gleich noch eine Neuerung. Mit dieser Ausgabe startet der 64'er-Hardware-Bestellservice.



Bild. Die Ziehung der Gewinner: Unsere Glücksfee Christiane war sich ihrer Verantwortung bewußt

Sie haben gewonnen!

Das Interesse an unserem Umfrage-Preisausschreiben der Ausgabe 6/85 sprengte alle Grenzen — mehr als 12000 Leser beteiligten sich. Gewonnen haben eigentlich alle, obwohl nicht jeder einen PC 10 mit nach Hause nehmen konnte.

Aber nun wollen wir endlich zu dem Punkt kommen, auf den Sie sicher schon lange gewartet haben: der Bekanntgabe der glücklichen Gewinner der Sachpreise. Unserer Glücksfee Christiane zitterten etwas die Hände (Bild), als sie den Gewinner des über 5500 Mark teuren PC 10 zog.

Der Gewinner des PC 10 ist:
Herr Martin Romisch, Frankfurt

Die Diskettenlaufwerke gewannen:
Brigitte Witzer, Ratingen; Georg Hüttinger, Stinglloh; Werner Krüger, Blaustein-Arnegg; Karsten Wenzel, Berlin; Rolf Beutner, Bohmte; Bernd Kattein, Sparneck; Holger Pietsch, Wunstorf; Hans-Peter Labude, Dreieich; Hans-Herrmann Arndt, Holzminden; Roland Rohe, Dortmund.

Je einen Buchgutschein im Wert von 50 Mark erhalten:
Detlef Metze, Andreas Schildbach, Soren Thade Petersen, Peter Hegelbach, Walter Stemmer, Peter Ackermann, Rolf Eckhardt, Christoph Balzer, Thomas Löper, Günter Meißner, Andreas Konrad, Werner Meyer, Christian Ruhl, W.L.M. Vermaegle, Christian Deissinger, Thomas Schäfer, Bernd Sauer, Peter Siegle, Daniel Reinert, Oliver Hertel, Udo Niehoegen, Volkmar Koch, Siegfried Pauer, Jens Thöming, Norbert Büchner, Bernd Bügner, André Kurzweg, Michael Weidel, Christian Meller, Georg Isak, Dirk Treppe, Zenek Przechacki, Hartmut Schulz, Sven-Eric Jordt, Helmut Nickl, Andreas Greß, Elmar Kellner, Christian Badry, Christian Patt, Matthias Eiser, Karl Wilhelm Vieten, Thorsten Müller, Thomas Meyer, Bernd Tophoven, Adolf Obenaus, Martin Horstkötter, Alexander Hild, Oliver Herrman, Herbert Niebuhr, Michael Spannuth.

Alle Gewinner werden innerhalb der nächsten Tage schriftlich benachrichtigt. Den Gewinner des Hauptpreises laden wir zu uns, nach Haar bei München in die Redaktion ein, um ihm seinen Hauptgewinn zu überreichen.

Herzlichen Glückwunsch allen Gewinnern. Aber auch ein ganz großes Dankeschön an alle die mitgemacht haben. Auch Ihre Teilnahme war nicht umsonst, denn das 64'er ist ein Magazin der Leser, und damit auf dem richtigen Weg. So gesehen gibt es eigentlich nur Gewinner!

Ihre 64'er Redaktion

Wir danken der Firma Commodore, Frankfurt, für die freundliche Stiftung des PC 10 und der Diskettenlaufwerke sowie dem Markt & Technik-Buchverlag, für die Buchpreise. (aw)

Programmier-Wettbewerb »Blockgrafik«

1000 Mark winken bei diesem Wettbewerb. Dabei ist die Aufgabe so einfach gestellt, daß wirklich jeder mitmachen kann, auch wenn er nur minimale Programmierkenntnisse besitzt.

Programmieren ist eine Faszination ganz besonderer Art. Doch immer wieder zeigt es sich, daß viele Leser Probleme haben, sich eine entsprechende Aufgabe zu stellen. Man freut sich regelrecht, wenn jemand zu einem kommt und fragt: »Ich habe ein Problem. Kann Dein Computer das lösen?« Na klar, denken wir, und schon ist man für einige Zeit von der Umwelt abgeschnitten, unterbrochen lediglich von einigen lebenserhaltenden Maßnahmen wie Essen, Trinken etc.

Und wenn das Problem nicht allzu anspruchsvoll ist, kommen wir eines Tages mit einem strahlenden Gesicht und der Lösung aus unserem Einsiedlerdasein zurück und präsentieren unsere geistige Leistung erwartungsvoll einem ewig kritischen Publikum, das keine Ahnung hat von den Mühen, die unser Programmierobjekt entstehen ließen. Doch wir wissen aus eigener Erfahrung, wie wertvoll jedes eingesandte Listing ist. Natürlich treffen wir eine Auswahl, wenn es auf die Suche nach einem veröffentlichbaren Programm geht. Doch mit jedem Programm, das Sie schreiben, gewinnen Sie. Und damit auch eine Aufgabe da ist, denken wir uns eine für Sie aus. Diesmal ist die Aufgabe besonders einfach.

Die Aufgabe

Bilder in Blockgrafik zu erstellen, verlangt mehr Kreativität als Programmierwissen. Anders als bei der hochauflösenden (HiRes) Grafik brauchen Sie weder den Grafik-Kurs gelesen zu haben noch andere, schwierige Literatur zu verarbeiten. Das einzige, das Sie sollen, ist, mit den Grafiksymbolen auf dem Tastenfeld Ihres Computers zu spielen, bis ein wirklich schönes oder interessantes Bild herauskommt. Ein paar PRINT-Zeilen, gespickt mit Grafiksymbolen und Farbe, garniert vielleicht mit etwas Text, reicht schon. Wollen Sie noch etwas mehr tun, können Sie noch Bewegung mit ins Spiel bringen. Sprites oder andere HiRes-Spielchen sind nicht erlaubt. Und übertreiben Sie bitte nicht bei der Länge des Programms. Wenn Sie Ihr Programm bis zum 30. Oktober 1985 eingeschickt haben, haben Sie alle Chancen, die 1000 Mark zu gewinnen, die auf den Sieger unseres Blockgrafik-Wettbewerbs warten. Schicken Sie Ihr Programm an

Markt & Technik Verlag Aktiengesellschaft,
64'er Redaktion,
Stichwort »Blockgrafik«,
Hans-Pinsel-Str. 2, 8013 Haar bei München □

Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen VC 20 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren VC 20/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats

500 Mark

wert.

Schreiben Sie uns, was Sie mit Ihrem Computer machen:

Redaktion 64'er, Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Einmal im Monat gibt es die SUPERCHANCE

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes, selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-)Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier) Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag und honorieren mit bis zu

2000 Mark für das Listing des Monats

Aus den besten Listings, die veröffentlicht werden, sucht die 64'er-Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in dem Beitrag »Wie schicke ich meine Programme ein?« in verschiedenen Ausgaben beschrieben.

Schicken Sie Ihr Listing an: Redaktion 64'er, Superchance: Listing des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.





Auf zu neuen Welten, Ausgabe 7/85, Seite 44

Leider treten in Verbindung mit manchen C 64 und dem IEEE-Interface Probleme auf. Statt des 4069-Bausteins sollte man dann einen 7405 oder 7406 (keine LS-Typen!) verwenden. Die Software ist in folgenden Punkten zu ändern: Ab Speicherstelle $FDDD folgende Bytefolge eingeben: $ AD A6 02 F0 00 A9 04 85 02 A9 25 8D 04 DC A9 40 4C F3 FD DC Verwenden Sie dazu am besten ein Monitor-Programm.

Die so geänderte Software wird wieder in ein 2764-EPROM gebrannt und in den Computer eingesetzt.

Ausgabe 7/85-MSE-Listing Terminalprogramm

Bei den von uns verschickten Seiten mit dem MSE-Listing des Programms »Terminalprogramm« ist bei der Montage ein Fehler aufgetaucht. In den Listings »Terminal« und »Editor« müssen die Zeilen $2081 bis $22b9 ausgetauscht werden. Das bedeutet, daß dieser Zeilenblock aus dem Listing »Terminal« in das Listing »Editor« geschrieben werden muß und umgekehrt.

Sound-Machine, Ausgabe 9/85, Seite 23

Im Listing 2, Noteneingabe, muß die Zeile 1140 »SYS 53027« lauten. Wird dies nicht korrigiert, stürzt der Computer beim Befehl »Save« ab.

Die Zeile 1890 muß ebenfalls »SYS 53027« heißen, der REM-Befehl entfällt.

Mousetrap, SH3 »Spiele«, Seite 57

In der »Hex-Eingabe-Routine« verhindert die Zeile 155 das ordnungsgemäße Funktionieren der Lade-Routine. Bitte ersetzen Sie die Zeile 155 duch »155 REM«.

Wir suchen die ersten 128er Profis

Der neue C 128 ist seit einigen Wochen im Handel. Vielleicht gehören Sie ja zu den ersten Besitzern dieses vielseitigen und leistungsfähigen neuen Computers und haben schon erste, tiefergehende Erfahrungen damit gesammelt oder interessante Programme dafür geschrieben?

In diesem Fall sollten Sie Ihre Erfahrungen nicht für sich behalten. Tragen Sie doch einfach mal zusammen, was Sie über den C 128 herausgefunden haben, das nicht im Handbuch steht. Wir werden alle guten Tips und Tricks zu diesem Computer sowie die besten Programme dafür im 64'er-Magazin veröffentlichen.

Bitte vermerken Sie bei allen Zusendungen, mit welcher Gerätekonfiguration Sie arbeiten und ob sich Ihr Beitrag auf den 128-Modus oder auf den CP/M-Modus bezieht. Beiträge für den C 64-Modus schicken Sie bitte nicht unter dem Stichwort C 128, sondern als normale C 64-Programme oder Tips ein. Bei Programmeinsendungen legen Sie bitte unbedingt eine Diskette/Kassette mit Ihrem Programm sowie eine möglichst ausführliche Beschreibung bei. Selbstverständlich werden alle abgedruckten Beiträge angemessen honoriert — es winken bis zu 2000 Mark, wenn Ihr Programm Listing des Monats wird.

Schicken Sie Ihre Tips und Programme an

Markt & Technik
Verlag Aktiengesellschaft
Redaktion 64'er
Kennwort: C 128
Hans-Pinsel-Str. 2
8013 Haar bei München

Inserentenverzeichnis



Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger (sc)
Leitender Redakteur: Albert Absmeier (aa)
Redakteure: ah = Achim Hübner, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, og = Markus Ohnesorg, tr = Thomas Röder, rg = Christian Rogge, aw = Arnd Wängler
Redaktionsassistenz: Yvonne Runge (202)
Fotografie: Janos Feitser/Jens Jancke, Titelfoto: Jens Jancke

Layout: Leo Eder (Ltg.), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-22 31 55/56, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenverkauf: Brigitta Fiebig (211)

Anzeigenverwaltung und Disposition: Michaela Hörl (171)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
Anzeigengrundpreise: 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
Anzeigen im Computer-Markt: Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (07 11) 64 83-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-1 19. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Verantwortlich: Für redaktionellen Teil: Michael Scharfenberger.
Für Anzeigen: Brigitta Fiebig.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 0 89-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.





Lyrik-Generator

Ein Basicprogramm macht Ihren Computer zum Dichter:
»Goldene Ränke zerschmettern eine Hoffnung —
Das Auge schreit —
Da immer wieder pulsierende Welten verwehen, stirbt das Universum.«

Eines von Millionen von Gedichten, die der C 64 in sekundenschnelle für Sie schreibt, wenn Sie dieses Programm haben. Wie Sie leicht an unserem Beispiel feststellen können, sind die Lyrik-Ergüsse des C 64 keine Aneinanderreihung von Wörtern, sondern syntaktisch und semantisch korrekte Sätze. Inwieweit allerdings diese Elektronik-Lyrik sinnvoll ist, entscheidet der Zufall. Lassen Sie sich überraschen. Wir sind allerdings davon überzeugt: Das ist die Anwendung des Monats.

Spiele-Trainer

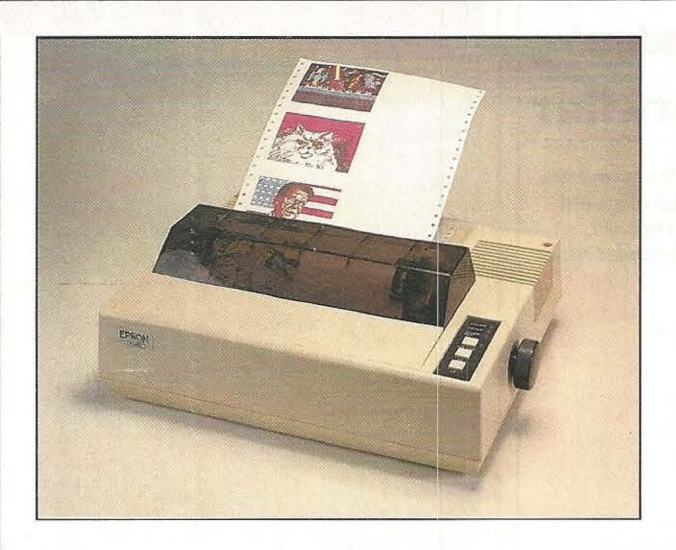
Kommen Sie bei ihrem Lieblings-Spiel nicht über das erste Bild hinaus? Werden Sie von ihrem Gegner stets niedergemetzelt, aufgefressen oder abgeschossen? Suchen Sie verzweifelt nach dem rettenden POKE, konnten ihn bisher aber nicht finden? Dann sollten Sie unseren Spiele-Trainer abtippen und vor dem Kampf laden. Die Angriffe der Gegner können Ihnen dann nichts mehr anhaben, denn Sie sind dagegen immun.

Schnellere 1541

Zurück zum Ursprung. Mit unserem Umbausatz verwandeln Sie die 1541 in ein 2031-Laufwerk mit Parallelübertragung, dem Vorgänger der 1541. Für weniger als 100 Mark können Sie sich einen Floppy-Speeder bauen, der die Geschwindigkeit fast aller Disketten-Optionen vervielfacht.

Außerdem ...

- 64'er Extra. Diesmal alle Kernalroutinen
- Die ersten Listings für den C 128
- Verbesserungen zu Hypra-Ass und SMON
- PC 128: Grafik im 80-Zeichen-Modus
- und wieder viele Tips und Tricks für den C 64 und C 128.



Druckertest

Vier neue Drucker testen wir für Sie: Epson JX-80, FX-85, Seikosha SP 1000 und den MPS 803 von Commodore. Lesen Sie, was den JX-80 außer Farbdruck noch auszeichnet und den FX-85 vom Spitzenreiter FX-80 unterscheidet. Daß Seikosha nicht nur preiswerte, sondern auch gute Drucker baut, beweist der SP 1000.

Alles über Joysticks

In der nächsten Ausgabe erfahren Sie alles über die »Steuerknüppel«, die zum unentbehrlichen Eingabegerät bei Spielen geworden sind. Neben einem Vergleichstest der wichtigsten Joysticks finden Sie eine Marktübersicht, die auch Trackballs und Lightpens berücksichtigt.



Der C 128 im 64'er-Magazin

Mittlerweile sind die ersten speziell an C 128 angepaßten CP/M-Programme erhältlich. Und das zu sensationellen Preisen von unter 200 Mark. Wir haben zwei der interessantesten Programme für Sie getestet: WordStar und Turbo Pascal. Beides sind professionelle Programme.

Computer-Grafik

Farbige Grafiken aus dem Wunderland der Mathematik. Sie erfahren, wie sich mit einfachen Mitteln komplizierte Computer-Grafiken erstellen lassen, die bis jetzt nicht, oder nur mit großem Aufwand, auf einem Heimcomputer möglich waren. Die fertigen Bilder lassen sich speichern und als Hardcopy mit Hi-Eddi drucken.

Textverarbeitung

Paper Clip erhebt den Anspruch, ein professionelles Textverarbeitungsprogramm zu sein. Wir haben diesen Anspruch getestet und sind auf einige bemerkenswerte Sonderfunktionen gestoßen. Kann sich Paper Clip, in Amerika lange Zeit Nummer eins, gegen Vizawrite, Textomat und Startexter behaupten? Lesen Sie unseren Test.

